



2

COMPUTER ASSISTED MEDICAL DIAGNOSIS (CAMD) SYSTEM
VERSION 1.0
TECHNICAL MANUAL

H. L. Ly

R. W. Dobbins

D. M. Pearsall



Report No. 92-25

93-04977



08 3 8 009

Approved for public release; distribution unlimited.

NAVAL HEALTH RESEARCH CENTER
P.O. BOX 85122
SAN DIEGO, CALIFORNIA 92186-5122

NAVAL MEDICAL RESEARCH AND DEVELOPMENT COMMAND
BETHESDA, MARYLAND



Computer Assisted Medical Diagnosis (CAMD) System
Version 1.0
Technical Manual

Prepared by:

Hoa L. Ly
Roy W. Dobbins
Dianna M. Pearsall

Naval Health Research Center
Medical Decisions Support Department
P.O. Box 85122
San Diego, CA 92186-5122

Accession For	
NTIS	CRA&I <input checked="" type="checkbox"/>
DTIC	TAB <input type="checkbox"/>
Unannounced <input type="checkbox"/>	
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Report No. TR 92-25 was supported by the Naval Medical Research and Development Command, Bethesda, MD, Department of the Navy, under Work Unit No. 63706N M0095.005-6103. The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, nor the U.S. Government.

SUMMARY

This technical manual contains the information on the program code, data elements, file structures needed to maintain the Computer Assisted Medical Diagnostic System. This documentation was created using the FOXDOC Version 2.0 program.

Table of Contents

Introduction	1
Section I. System Summary.	1
Section II. Menu Summary.	1
Section III. Screen Summary.	4
section IV. Report Form File Summary	21
Section V. Data Dictionary	22
Database Structure Summary	22
Database Field Summary	28
Index Parameter Summary	30
Section VI. Tree Diagram.	34
Section VII. Procedure and Function Summary.	51
Section VIII. Program Source Code	75

Introduction. This document was created using the FOXDOC Version 2.0 program to generate the technical documentation for the Computer Assisted Medical Diagnosis System (CAMD) system. The documentation is separated into eight sections (1) System Summary, (2) Menu Summary, (3) Screen Summary, (4) Report Form File Summary, (5) Data Dictionary, (6) Tree Diagram, (7) Procedure Summary, and (8) Program Source Code.

Section I. System Summary. See the tree diagram for programs, procedures, functions and format files.

This system has:

- 17695 lines of code
 - 1 program file
 - 59 procedure files
- 301 procedures and functions
- 23 databases
 - 1 structural index file
- 31 index files
 - 1 report form
- 2 menu files
- 29 screen files
- 4 other files
- 2 cross-referenced tokens

Section II. Menu Summary. The following description lists the menu options used to drive the system. Two menu files in CAMD system are CAMD.MNX and CAMDTR.MNX.

CAMD.MNX is main menu to driven the system. Last updated: 10/07/92 at 11:51

<u>OPTION NAME</u>	<u>KEYS\FUNCTION CALLED</u>
System	ALT+S (Submenu _CSYSTEM)
Help...	F1 _MST_HELP

Change Password	do pwdedt
Backup	do _mhit with "backup"
Restore	do _mhit with "restore"

Filer	_MST_FILER
Calculator	_MST_CALCUL
Calendar/Diary	_MST_DIARY
Puzzle	_MST_PUZZL
File	ALT+F (Submenu _CFILE)
Reindex	DO reindex

Quit	DO _quit
Edit	ALT+E (Submenu _CEDIT)
Undo	CTRL+U _MED_UNDO
Redo	CTRL+R _MED_REDO

Cut	CTRL+X _MED_CUT
Copy	CTRL+C _MED_COPY
Paste	CTRL+V _MED_PASTE
Clear	_MED_CLEAR

Select All	CTRL+A _MED_SLCTA

Goto Line...	_MED_GOTO
Find...	CTRL+F _MED_FIND
Find Again	CTRL+G _MED_FINDA

Replace All	<u>_MED_REPLA</u>

Preferences...	<u>_MED_PREF</u>
Diagnosis	<u>ALT+D</u> (Submenu DIAGNOSIS)
Register Patient...	DO <u>_mhit</u> WITH " <u>_m_pat</u> "
Encounter...	DO <u>_mhit</u> WITH " <u>_m_newenc</u> "
SF600 Report	DO <u>_mhit</u> WITH " <u>_m_sf600</u> "

Disease/Treatment Lookup	DO <u>_mhit</u> WITH " <u>_m_lookup</u> "

Encounter Training Mode	(procedure)
Window	<u>ALT+W</u> (Submenu <u>_CWINDOW</u>)
Move	<u>CTRL+F7</u> <u>_MWI_MOVE</u>
Size	<u>CTRL+F8</u> <u>_MWI_SIZE</u>
Zoom ^X	<u>CTRL+F10</u> <u>_MWI_ZOOM</u>
Zoom ^Y	<u>CTRL+F9</u> <u>_MWI_MIN</u>
Cycle	<u>CTRL+F1</u> <u>_MWI_ROTAT</u>
Color...	do colorset.spr

Command	<u>CTRL+F2</u> <u>_MWI_CMD</u>
Debug	<u>_MWI_DEBUG</u>
Trace	<u>_MWI_TRACE</u>
View	<u>_MWI_VIEW</u>

CAMDTR.MNX is a menu to use for training. Last updated: 09/28/92 at 14:19

<u>OPTION NAME</u>	<u>KEYS\FUNCTION CALLED</u>
System	<u>ALT+S</u> (Submenu <u>_CSYSTEM</u>)
Help...	<u>F1</u> <u>_MST_HELP</u>

Change Password	do pwdedt
Backup	do <u>_mhit</u> with "backup"
Restore	do <u>_mhit</u> with "restore"

Filer	<u>_MST_FILER</u>
Calculator	<u>_MST_CALCUL</u>
Calendar/Diary	<u>_MST_DIARY</u>
Puzzle	<u>_MST_PUZZL</u>
File	<u>ALT+F</u> (Submenu <u>_CFILE</u>)
Reindex	DO reindex

Quit	DO <u>_quit</u>
Edit	<u>ALT+E</u> (Submenu <u>_CEDIT</u>)
Undo	<u>CTRL+U</u> <u>_MED_UNDO</u>
Redo	<u>CTRL+R</u> <u>_MED_REDO</u>

Cut	<u>CTRL+X</u> <u>_MED_CUT</u>
Copy	<u>CTRL+C</u> <u>_MED_COPY</u>
Paste	<u>CTRL+V</u> <u>_MED_PASTE</u>
Clear	<u>_MED_CLEAR</u>

Select All	<u>_MED_SP200</u>

Goto Line...	<u>_MED_GOTO</u>
Find...	<u>CTRL+F</u> <u>_MED_FIND</u>
Find Again	<u>CTRL+G</u> <u>_MED_FINDA</u>
Replace And Find Again	<u>CTRL+E</u> <u>_MED_REPL</u>
Replace All	<u>_MED_REPLA</u>

Preferences...	<u>_MED_PREF</u>
Diagnosis	<u>ALT+D</u> (Submenu DIAGNOSIS)
Register Patient...	DO <u>_mhit</u> WITH " <u>_m_pat</u> "

Encounter...	DO _mhit WITH "m_newenc"
SF600 Report	DO _mhit WITH "m_sf600"

Disease/Treatment Lookup	DO _mhit WITH "m_lookup"

Encounter Training Mode	DO _mhit WITH "m_training"
Window	ALT+W (Submenu_CWINDOW)
Move	CTRL+F7 _MWI_MOVE
Size	CTRL+F8 _MWI_SIZE
Zoom ^X	CTRL+F10 _MWI_ZOOM
Zoom ^Y	CTRL+F9 _MWI_MIN
Cycle	CTRL+F1 _MWI_ROTAT
Color...	_MWI_COLOR

Command	CTRL+F2 _MWI_CMD
Debug	_MWI_DEBUG
Trace	_MWI_TRACE
View	_MWI_VIEW
TRAINING MODE	

Section III. Screen Summary. Twenty eight (28) screen files were used as input templates in the system; W14.SCX, W10.SCX, W7.SCX, W6.SCX, W6C.SCX, W6B.SCX, W11.SCX, W12.SCX, W2.SCX, W9.SCX, W5.SCX, W3.SCX, W13.SCX, WP3.SCX, WP4.SCX, WP5.SCX, WP6.SCX, WP7.SCX, WP8.SCX, WP9.SCX, W12A.SCX, W12B.SCX, W12C.SCX, W12E.SCX, BACKUP.SCX, RESTORE.SCX, W7A.SCX, COLORSET.SCX.

W14.SCX Last updated: 02/12/92 at 10:50

```

0
1
2 1: mchoice.....▲
3 .....▲
4 .....▲
5 .....▲
6 .....▲
7 .....▲
8 .....▲
9 .....▼
10
11      < OK > <Cancel>

```

Window name: W_list
Coordinates: FROM 0,0 TO 0,51
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mchoice	List	"@t"
2: Mbuttons	Push button	"@*HT OK;\?Cancel"

W10.SCX Last updated: 06/03/92 at 13:39

```

Confirm
0
1 1: .....
2 .....
3 .....
4 .....
5
6      < OK > <Cancel>

```

Name	Type	Picture
1: m.prompt	Field	
2: mbuttons	Push button	"@*HT OK;Cancel"

W7.SCX

Last updated: 05/28/92 at 8:41

Diseases on File

0
1 1: mchoice.....
2
3
4
5
6
7
8
9
10
11

12 <Select> < View > <Cancel>

Window name: W disease

Coordinates: FROM 0,0 TO 0,38

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mchoice	List	"@&T"
2: mbuttons	Push button	"@*HT Select;View;\?Cancel"

W6.SCX

Last updated: 04/02/92 at 17:27

Numeric Data Entry

0
1 2:
2
3
4
5 Range 5: mlo.... to 3: mhi....
6
7
8 1: mval... 4: munits.....
9
10
11
12 < OK > <Cancel>

Window name: W_num

Coordinates: FROM 0,0 TO 0,54

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mval	Field	"@KZ"
2: mprompt	Field	"@T"
3: mhi	Field	"@BZ"
4: munits	Field	"@T"
5: mlo	Field	"@BZ"
6: mbuttons	Push button	"@*HT OK;\?Cancel"

W6C.SCX

Last updated: 04/02/92 at 17:29

Logical Data Entry

```
0
1 2: .....
2 .....
3 .....
4
5
6      ( = ) Yes
7      (   ) No
8
9
10
11      < OK >      <Cancel>
```

Window name: W check
Coordinates: FROM 0,0 TO 5,53
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mbuttons	Push button	"@*HT OK;\?Cancel"
2: mprompt	Field	"@T"
3: mchoice	Radio button	"@*RVN Yes;No"

W6B.SCX

Last updated: 04/02/92 at 17:28

Single Choice Data Entry

```
0
1 2: .....
2 .....
3 .....
4
5
6      3: mchoice.....
7 .....
8 .....
9 .....
10 .....
11 .....
12 .....
13
14      < OK >      <Cancel>
```

Window name: W check
Coordinates: FROM 0,0 TO 7,62
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mbuttons	Push button	"@*HT OK;\?Cancel"
2: mprompt	Field	"@T"
3: mchoice	List	"@&T"

Last updated: 06/09/92 at 15:20

```

0
1 Patient 3: mkey...
2
3 Date 4: mdate..

```

```

6
7 1: mchoice.....▲
8 .....◆
9 .....
10 .....
11 .....
12 .....
13 .....▼
14

```

15
16 <Resume> < New > <Cancel>

```
Window name: W_pend
Coordinates: FROM 0,0 TO 0,58
Window options: FLOAT CLOSE MINIMIZE SHADOW
```

Name	Type	Picture
1: mchoice	List	"@&N"
2: mbuttons	Push button	"@*HT Resume;New;\?Cancel"
3: mkey	Field	
4: mdate	Field	

W12.SCX

Last updated: 10/13/92 at 14:16

Patient Encounter

```

0 Date 1: me_da Time 2: me Status 3: m Provider 4: me_providr...
1 Patient Name 5: me_lastnam..... First 6: me_firstna.....
2 SSN 7: me_ssn.. Sex 8: me_ Age 9:
3
4 <Complaint > 11: me_area..... <Diagnosis> 15: me_diagnos.....
5 <Symptom> 22 <Assist> 22
6
7 13: mcondL.....
8 .....
9 .....
10 .....
11 .....
12 .....
13
14 <Treatment > 21
15 20: me_treat.....
16 .....
17 .....
18
19 <Print > «Save» <Cancel>

```

Window name: W_encount

Coordinates: FROM 0,0 TO 0,79

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: me_dateenc	Field	
2: me_timeenc	Field	
3: me_state	Field	
4: me_providr	Field	
5: me_lastnam	Field	
6: me_firstna	Field	"@!"
7: me_ssn	Field	"@R 999-99-9999"
8: me_sex	Field	"@!"
9: me_age	Field	
10: mckcompl	Push button	"@*HN <Complaint"
11: me_area	Field	
12: mckhandp	Push button	"@*HN <Symptom"
13: mcondL	List	"@&N"
14: mckdiag	Push button	"@*HN <Diagnosis"
15: me_diagnos	Field	
16: mckassist	Push button	"@*HN <Assist"
17: mdiagL	List	"@&N"
18: mcktreat	Push button	"@*HN <Treatment"
19: chr(30)	Field	
20: me_treat	Field	
21: mbuttons	Push button	"@*HT <Print; ?Sa <ve;Ca <ncel"
22: chr(31)	Field	
23: chr(31)	Field	
24: chr(30)	Field	
25: chr(31)	Field	
26: chr(30)	Field	

W2.SCX

Last updated: 06/15/92 at 15:01

Print Encounter

```
0
1 Print Encounter to:      (*) File
2                        ( ) Printer
3
4
5
6 Filename: 1: mprintfile.....
7
8
9
10      < OK >      <Cancel>
```

Window name: W_prn
Coordinates: FROM 0,0 TO 0,42
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mprintfile	Field	
2: mbuttons	Push button	"@*HT OK;\?Cancel"
3: mchoice	Radio button	"@*RVN File;Printer"

W9.SCX

Last updated: 04/02/92 at 17:33

Warning

```
0
1 1: .....
2 .....
3 .....
4
5      <OK>
```

Window name: W_popmsg
Coordinates: FROM 0,0 TO 0,35
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: errstr	Field	
2: mbuttons	Push button	"@*HT OK"

W5.SCX

Last updated: 05/28/92 at 8:17

Complaints

0	
1	2: mchoice.....▲
2◆
3
4
5
6
7
8▼

<Select>
<Cancel>

Window name: W_area
Coordinates: FROM 0,0 TO 8,45
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mbuttons	Push button	"@*VT Select;\?Cancel"
2: mchoice	List	"@&T"

W3.SCX

Last updated: 05/28/92 at 8:15

Patients on File

0	
1	1: mchoice.....▲
2◆
3
4
5
6
7
8
9
10▼
11	
12	<Select> <Cancel>

Window name: W_patlist
Coordinates: FROM 0,0 TO 0,36
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mchoice	List	"@&T"
2: mbuttons	Push button	"@*HT Select;\?Cancel"

W13.SCX

Last updated: 02/12/92 at 10:21

Undo Options

```

0
1
2  1: mchoice.....▲
3  .....
4  .....
5  .....
6  .....
7  .....
8
9  < OK > <Cancel>

```

Window name: W_undo
 Coordinates: FROM 0,0 TO 0,31
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mchoice	List	"@&T"
2: Mbuttons	Push button	"@*HT OK; ?Cancel"

WP3.SCX

Last updated: 08/03/92 at 16:03

Patient in File

```

0
1  1: mchoice.....▲
2  .....
3  .....
4  .....
5  .....
6  .....
7  .....
8  .....
9  .....
10 .....
11 .....
12 <Select> < Edit > <Cancel>

```

Window name: Wp3
 Coordinates: FROM 0,0 TO 0,40
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mchoice	List	"@&T"
2: mbuttons	Push button	"@*HT <Select; <Edit; ? <Cancel"

WP4.SCX

Last updated: 06/14/92 at 17:37

Encounter to Report

```

0
1  Social Security: 1: mssn.....
2
3  Last Name:      2: mlast.....
4
5  First Name:..   3: mfirst.....
6
7  Encounter date: 4: mdateen
8
9      " OK "    <Browse>    <Cancel>

```

Window name: Wp4
Coordinates: FROM 0,0 TO 0,55
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mssn	Field	"@R 999-99-9999"
2: mlast	Field	
3: mfirst	Field	
4: mdateenc	Field	
5: mbuttons	Push button	"@*HT 1 <OK; <Browse; <Cancel"

WP5.SCX

Last updated: 06/14/92 at 18:30

```

0 Patient  3: mkey.....
1
2 Date     4: mdate..
3
4
5 1: mchoice.....▲
6 .....▼
7 .....
8 .....
9 .....
10 .....
11 .....
12 .....
13
14      < OK >    <Cancel>

```

Window name: Wp4
Coordinates: FROM 0,0 TO 0,58
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mchoice	List	"@&N"
2: mbuttons	Push button	"@*HT <OK; ?Cancel"
3: mkey	Field	
4: mdate	Field	

Report Editing

```

0
1 1: mstr.....▲
2 .....♦
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
10 .....
11 .....
12 .....
13 .....
14 .....
15 .....▼
16
17          <Print >    < Save >    <Cancel>

```

Window name: Wp6
 Coordinates: FROM 0,0 TO 0,76
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mstr	Field	
2: mbuttons	Push button	"@*HT <Print; <Save; <Cancel"

WP7.SCX

Last updated: 06/16/92 at 13:00

Print Option

```

0
1 Print Encounter to:  ( = ) File
2                      ( ) Printer
3
4
5
6 Filename: 1: mprintfile.....
7
8
9
10      < OK >      <Cancel>

```

Window name: Wp7
Coordinates: FROM 0,0 TO 0,42
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mprintfile	Field	
2: mbuttons	Push button	"@*HT OK;\?Cancel"
3: mchoice	Radio button	"@*RVN File;Printer"

WP8.SCX

Last updated: 06/16/92 at 9:32

Patient in File

```

0
1 1: mchoice.....▲
2 .....◆
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
10 .....▼
11
12      <Select>      <Cancel>

```

Window name: Wp3
Coordinates: FROM 0,0 TO 0,40
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mchoice	List	"@&T"
2: mbuttons	Push button	"@*HT \<Select;\?<Cancel"

WP9.SCX

Last updated: 06/24/92 at 10:32

```

0 1: .....
1 .....
2 .....
3 .....

```

Window name: Wp9
 Coordinates: FROM 1,0 TO 4,76
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mtext	Field	

W12A.SCX

Last updated: 07/06/92 at 10:38

```

0                                     <Symptom>                                     34
1
2  1: mcondL.....
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
10 .....
11 .....
12 .....
13 .....
14 .....
15 .....
16 .....
17 .....
18 .....
19 .....
20

```

Window name: W_handp
 Coordinates: FROM 0,0 TO 0,79
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mcondL	List	"@&N"
2: mckhandp	Push button	"@*HN \<Symptom"
3: chr(31)	Field	
4: chr(30)	Field	

	<Assist>	43
0		
1		
2	1: mdiagL.....	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19		
20		

Window name: W_assist

Coordinates: FROM 0,0 TO 0,79

Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mdiagL	List	"@&N"
2: mckassist	Push button	"@*HN \<Assist"
3: chr(30)	Field	
4: chr(31)	Field	

W12C.SCX

Last updated: 07/06/92 at 10:41

Treatment

	<Treatment >	32
0		
1		
2	1: me_treat.....	▲
3	◆
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	▼
19		

Window name: W treat
 Coordinates: FROM 0,0 TO 0,79
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: me_treat	Field	
2: chr(30)	Field	
3: chr(31)	Field	
4: mcktreat	Push button	"@*HN \<Treatment"

W12E.SCX

Last updated: 07/20/92 at 12:20

0 1:
 1

Window name: W warning
 Coordinates: FROM 23,0 TO 24,79
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mprivateact	Field	77

```

0  2: message.....
1
2
3  5: mfile.....▲
4  .....
5  .....
6  .....
7  .....
8  .....
9  .....
10 .....
11 .....▼
12
13
14 Backup file name:.6: mfname.....

```

To drive: 1: mdrive.....

Directory: 3: mpath.....

« Back »

<Cancel>

Window name: W_backup
 Coordinates: FROM 0,0 TO 0,60
 Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mdrive	Popup	"@^ "
2: message	Field	
3: mpath	Popup	"@^ "
4: maction	Push button	"@*VT <Backup;<Cancel"
5: mfile	List	"@&N"
6: mfname	Field	

0 2: message.....

1

2

3 5: mfile.....▲

4▼

5▼

6▼

7▼

8▼

9▼

10▼

11▼

12

13

From drive: 1: mdrive.....

Directory: 3: mpath.....

«Restore»

<Cancel >

14 Restore file name:.6: mfname.....

Window name: W_restore
Coordinates: FROM 0,0 TO 0,60
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mdrive	Popup	"@^ "
2: message	Field	
3: mpath	Popup	"@^ "
4: maction	Push button	"@*VT <Restore;<Cancel"
5: mfile	List	"@&N"
6: mfname	Field	

W7A.SCX

Last updated: 08/03/92 at 16:04

Disease/Treatment Reference

```

0
1  1: mchoice.....▲
2  .....
3  .....
4  .....
5  .....
6  .....
7  .....
8  .....
9  .....
10 .....
11 .....
12 .....
13 .....
14 .....▼
15

```

« View » < Quit >

Window name: W dislook
Coordinates: FROM 0,0 TO 0,39
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: mchoice	List	"@&T"
2: mbuttons	Push button	"@*HT <View> <Quit"

COLORSET.SCX

Last updated: 10/07/92 at 11:39

```

0
1  1: coloridx.....▲
2  .....
3  .....
4  .....
5  .....
6  .....
7  .....
8  .....
9  .....▼
10
11

```

< Load > <Cancel>

Window name: W color
Coordinates: FROM 0,0 TO 0,31
Window options: FLOAT CLOSE MINIMIZE SHADOW

Name	Type	Picture
1: m.coloridx	List	"@&N"
2: m.action	Push button	"@*HT <Load> <Cancel"

section IV. Report Form File Summary. The only one report forms was created in CAMD system; SF600.FRX. Last updated: 07/09/92 at 16:02

Report Contents				
No.	Field	Length	Row	Column
1	PAGENO	4	0	73
2	mdateenc	8	11	0
3	mtext	65	11	11
4	mtimeenc	8	12	0
5	msocsec	11	53	0
6	alltrim(mlastname)+ ", " + alltrim(mfirstname)+ " " + alltrim(mmiddlename)	28	53	14
7	mdatebirth	12	53	45
8	msex	6	53	60
9	mraterank	10	53	69

		152		

Database and Program References

There is no an associated database

Used by: SFPRINT

(procedure in C:\CAMD\WORK\U.PRG)

Section V. Data Dictionary. There are eighteen (18) database files in the CAMD system:

SECTION.DBF	--	SF600 report section.
REPORT.DBF	--	SF600 report of each CAMD system area.
SF600TEX.DBF	--	The natural language of sign and symptom, use to generate SF600 report.
AREA.DBF	--	Diagnosis area of the CAMD System.
DICT.DBF	--	Sign & Symptom Question.
VAL.DBF	--	Range of Numeric type
ENUM.DBF	--	The response to the signs and symptoms question
DISEASE.DBF	--	Disease description and Treatment protocol.
PREMISE.DBF	--	Rule premise file.
ACTION.DBF	--	Rule action file.
RULE.DBF	--	Rules database file.
SYSTEM.DBF	--	CAMD system configure, greeting message.
PATIENT.DBF	--	Patient demographic database.
ENCOUNTR.DBF	--	Patient encounter record.
SYMPTOMS.DBF	--	Sign and Symptoms input record.
DIAGNOSE.DBF	--	Diagnosis output by CAMD system.
BACKUP.DBF	--	List of database file to be backup.
KEYLOG.DBF	--	Action record of user using CAMD system.

Database Structure Summary

Structure for database : SECTION.DBF

Number of data records : 81

Last updated : 09/23/92 at 12:12

Field	Field name	Type	Width	Dec	Start	End
1	SECTION	Numeric	3		1	3
2	OBJECT	Character	1		4	4
3	ID	Numeric	5		5	9
4	SENTENCE	Numeric	3		10	12
** Total **			13			

Used by: SFSYMP (procedure in DEMO.PRG)
: REINDEX (procedure in U.PRG)

Structure for database : REPORT.DBF

Number of data records : 59

Last updated : 09/23/92 at 12:13

Field	Field name	Type	Width	Dec	Start	End
1	SECTION	Numeric	3		1	3
2	AREA	Numeric	4		4	7
** Total **			8			

Used by: STRNATURAL (procedure in DEMO.PRG)
: SFSYMP (procedure in DEMO.PRG)
: REINDEX (procedure in U.PRG)

Structure for database : SF600TEX.DBF
 Number of data records : 81
 Last updated : 09/23/92 at 12:12

Field	Field name	Type	Width	Dec	Start	End
1	SENTENCE	Numeric	3		1	3
2	OBJECT	Character	1		4	4
3	ID	Numeric	5		5	9
4	STARTTEXT	Character	60		10	69
5	ENDTEXT	Character	60		70	129
6	DEFAULT	Character	120		130	249
7	NOPREVSEN	Character	60		250	309
** Total **			310			

Used by: SFSYMP (procedure in DEMO.PRG)
 : SFPRINT (procedure in U.PRG)
 : REINDEX (procedure in U.PRG)

Structure for database : AREA.DBF
 Number of data records : 2
 Last updated : 08/03/92 at 11:16

Field	Field name	Type	Width	Dec	Start	End
1	AREA	Numeric	4		1	4
2	NAME	Character	30		5	34
3	RULEBASE	Character	10		35	44
4	SIGNON	Character	30		45	74
5	START	Character	30		75	104
6	FINISH	Character	30		105	134
7	THRESHOLD	Numeric	6	2	135	140
8	PROBABLE	Numeric	6	2	141	146
9	LIKELY	Numeric	6	2	147	152
10	RULES	Numeric	4		153	156
** Total **			157			

Used by: GETEXAREA() (function in U.PRG)
 : GETRULEBASE() (function in U.PRG)
 : AREACOUNT() (function in U.PRG)
 : AREANAMES() (function in U.PRG)
 : ENCNAMES() (function in U.PRG)
 : SFENCNAMES (procedure in U.PRG)
 : OPENFILE (procedure in QUESTION.PRG)

Structure for database : DICT.DBF
 Number of data records : 80
 Last updated : 09/02/92 at 14:36

Field	Field name	Type	Width	Dec	Start	End
1	ID	Numeric	4		1	4
2	NAME	Character	20		5	24
3	DATATYPE	Character	1		25	25
4	ASKABLE	Logical	1		26	26
5	QUESTION	Character	100		27	126
6	PROMPT	Character	80		127	206
** Total **			207			

Used by: DICTCOUNT() (function in U.PRG)
 : OPENDICT (procedure in U.PRG)
 : REINDEX (procedure in U.PRG)
 : OPENFILE (procedure in QUESTION.PRG)

Structure for database : VAL.DBF Alias: DICTVAL
 Number of data records : 11
 Last updated : 07/23/92 at 8:32

Field	Field name	Type	Width	Dec	Start	End
1	ID	Numeric	4		1	4
2	WIDTH	Numeric	2		5	6
3	DEC	Numeric	1		7	7
4	LO	Numeric	9	2	8	16
5	HI	Numeric	9	2	17	25
6	UNITS	Character	10		26	35
** Total **			36			

Used by: OPENDICT (procedure in U.PRG)
 : REINDEX (procedure in U.PRG)
 : OPENFILE (procedure in QUESTION.PRG)

Structure for database : ENUM.DBF
 Number of data records : 251
 Last updated : 07/23/92 at 11:09

Field	Field name	Type	Width	Dec	Start	End
1	ID	Numeric	4		1	4
2	ORD	Numeric	2		5	6
3	MUTEX	Character	1		7	7
4	ENUMERATE	Character	80		8	87
5	REPORT	Character	80		88	167
** Total **			168			

Used by: OPENDICT (procedure in U.PRG)
 : REINDEX (procedure in U.PRG)
 : OPENFILE (procedure in QUESTION.PRG)

Structure for database : DISEASE.DBF
 Number of data records : 20
 Last updated : 10/08/92 at 11:58

Field	Field name	Type	Width	Dec	Start	End
1	NAME	Character	60		1	60
2	DESCRIPT	Memo	10		61	70
3	ID	Numeric	5		71	75
4	SOURCE	Character	1		76	76
5	XID	Numeric	5		77	81
6	AREA	Numeric	4		82	85
7	TREATMENT	Memo	10		86	95
8	BRIEF	Memo	10		96	105
** Total **			106			

This database is associated with the memo file: DISEASE.FPT

Used by: DISEASENAM() (function in U.PRG)
 : DISEASEID() (function in U.PRG)
 : DISEASECNT() (function in U.PRG)
 : DISNAMES() (function in U.PRG)
 : DISEASEGET (procedure in U.PRG)
 : DISEASEPUT (procedure in U.PRG)
 : REINDEX (procedure in U.PRG)

Structure for database : PREMISE.DBF

Number of data records : 372

Last updated : 07/23/92 at 11:22

Field	Field name	Type	Width	Dec	Start	End
1	CLAUSE	Numeric	5		1	5
2	OP	Character	2		6	7
3	OBJECT	Character	1		8	8
4	ID	Numeric	5		9	13
5	VAL	Character	30		14	43
** Total **			44			

Used by: OPENRULE (procedure in U.PRG)
: REINDEX (procedure in U.PRG)
: OPENFILE (procedure in QUESTION.PRG)

Structure for database : ACTION.DBF

Number of data records : 1779

Last updated : 06/09/92 at 15:01

Field	Field name	Type	Width	Dec	Start	End
1	CLAUSE	Numeric	5		1	5
2	OP	Character	2		6	7
3	OBJECT	Character	1		8	8
4	ID	Numeric	5		9	13
5	VAL	Character	30		14	43
** Total **			44			

Used by: OPENRULE (procedure in U.PRG)
: REINDEX (procedure in U.PRG)
: OPENFILE (procedure in QUESTION.PRG)

Structure for database : RULE.DBF

Number of data records : 330

Last updated : 08/04/92 at 10:27

Field	Field name	Type	Width	Dec	Start	End
1	RULE	Numeric	5		1	5
2	AREA	Numeric	4		6	9
3	SALIENCE	Numeric	3		10	12
4	PREMISE	Numeric	5		13	17
5	ACTION	Numeric	5		18	22
6	TEXT	Memo	10		23	32
** Total **			33			

This database is associated with the memo file: RULE.FPT

Used by: OPENRULE (procedure in U.PRG)
: REINDEX (procedure in U.PRG)
: OPENFILE (procedure in QUESTION.PRG)

Structure for database : SYSTEM.DBF

Number of data records : 1

Last updated : 10/13/92 at 9:58

Field	Field name	Type	Width	Dec	Start	End
1	PASSWORD	Character	10		1	10
2	PATIENTS	Numeric	4		11	14
3	PROVIDERS	Numeric	4		15	18
4	ENCOUNTERS	Numeric	4		19	22
5	STATION	Character	30		23	52
6	DEPARTMENT	Character	30		53	82
7	SYSWARNING	Memo	10		83	92
8	SYSDISCL	Memo	10		93	102
9	PRIVACYACT	Memo	10		103	112
10	DEBUG	Logical	1		113	113
11	CONF	Logical	1		114	114
12	SHOWRULE	Logical	1		115	115
13	NOVICE	Logical	1		116	116
14	EDITHELP	Logical	1		117	117
** Total **			118			

This database is associated with the memo file: SYSTEM.FPT

Used by: PWDEDT.PRG

: NEXTENCNT()	(function in U.PRG)
: GETSTATION()	(function in U.PRG)
: GETOPTS	(procedure in U.PRG)
: NEXTPAT	(procedure in U.PRG)
: NENCNT	(procedure in U.PRG)
: NEXTUSER	(procedure in U.PRG)
: GETSYSCODE()	(function in SECHECK.PRG)
: OPENSAMS	(procedure in U.PRG)

Structure for database : PATIENT.DBF

Number of data records : 2

Last updated : 10/09/92 at 14:11

Field	Field name	Type	Width	Dec	Start	End
1	ID	Numeric	4		1	4
2	SOCSEC	Character	9		5	13
3	LASTNAME	Character	30		14	43
4	FIRSTNAME	Character	30		44	73
5	MIDDLEINIT	Character	1		74	74
6	DATEBIRTH	Date	8		75	82
7	SEX	Character	1		83	83
8	RATERANK	Character	20		84	103
** Total **			104			

Used by: REINDEX

(procedure in U.PRG)

Structure for database : ENCOUNTR.DBF

Number of data records : 1

Last updated : 10/13/92 at 10:19

Field	Field name	Type	Width	Dec	Start	End
1	ENCOUNTRID	Numeric	5		1	5
2	PATIENT	Numeric	4		6	9
3	DATE	Date	8		10	17
4	TIME	Character	8		18	25
5	PROVIDER	Numeric	4		26	29
6	STATION	Character	30		30	59
7	DEPARTMENT	Character	30		60	89
8	AREA	Numeric	5		90	94
9	DIAGNOSE	Numeric	5		95	99
10	COMPLAINT	Character	30		100	129
11	TREATMENT	Memo	10		130	139
12	REPORT	Memo	10		140	149
13	STATE	Character	1		150	150
** Total **			151			

This database is associated with the memo file: ENCOUNTR.FPT

Used by: REINDEX (procedure in U.PRG)

Structure for database : SYMPTOMS.DBF

Number of data records : 2

Last updated : 10/13/92 at 10:19

Field	Field name	Type	Width	Dec	Start	End
1	ID	Numeric	5		1	5
2	SYMPTOM	Numeric	4		6	9
3	VALUE	Character	10		10	19
** Total **			20			

Used by: REINDEX (procedure in U.PRG)

Structure for database : DIAGNOSE.DBF

Number of data records : 0

Last updated : 09/03/92 at 12:45

Field	Field name	Type	Width	Dec	Start	End
1	ID	Numeric	5		1	5
2	DISEASE	Numeric	5		6	10
3	PROBABILTY	Numeric	6	2	11	16
** Total **			17			

Used by: REINDEX (procedure in U.PRG)

Structure for database : BACKUP.DBF

Number of data records : 17

Last updated : 09/03/92 at 12:44

Field	Field name	Type	Width	Dec	Start	End
1	FILE NAME	Character	12		1	12
** Total **			13			

This database appears to be associated with multiple index file(s):

: BACKUP.CDX

: PKUNZIP

(procedure in RESTORE.SPR)

Structure for database : KEYLOG.DBF
 Number of data records : 34
 Last updated : 10/13/92 at 10:29

Field	Field name	Type	Width	Dec	Start	End
1	ACTION	Character	60		1	60
2	DATE	Date	8		61	68
3	TIME	Character	8		69	76
** Total **			77			

Used by: KEYLOG.PRG

Structure for database : COLORSET.DBF
 Number of data records : 7
 Last updated : 10/07/92 at 11:52

Field	Field name	Type	Width	Dec	Start	End
1	COLOR_SET	Character	30		1	30
** Total **			31			

Used by: COLORSET.SPR

Database Field Summary: A listing of alphabetical database's field name was used in the CAMD system.

Field Name	Type	Len	Dec	Database
ACTION	C	60	0	KEYLOG.DBF
ACTION	N	5	0	RULE.DBF
AREA	N	4	0	RULE.DBF
AREA	N	5	0	REPORT.DBF
AREA	N	4	0	ENCOUNTR.DBF
AREA	N	4	0	AREA.DBF
ASKABLE	L	1	0	DISEASE.DBF
BRIEF	M	10	0	DICT.DBF
CLAUSE	N	5	0	DISEASE.DBF
COLOR_SET	C	30	0	ACTION.DBF
COMPLAIN	C	30	0	PREMISE.DBF
CONF	L	1	0	COLORSET.DBF
DATATYPE	C	1	0	ENCOUNTR.DBF
DATE	D	8	0	SYSTEM.DBF
DATEBIRTH	D	8	0	DICT.DBF
DEBUG	L	1	0	KEYLOG.DBF
DEC	N	1	0	ENCOUNTR.DBF
DEFAULT	C	120	0	PATIENT.DBF
DEPARTMENT	C	30	0	SYSTEM.DBF
DESCRIPT	M	10	0	VAL.DBF
DIAGNOSE	N	5	0	SF600TEX.DBF
DISEASE	N	5	0	ENCOUNTR.DBF
EDITHELP	L	1	0	DIAGNOSE.DBF
ENCOUNTERS	N	4	0	SYSTEM.DBF
ENCOUNTRID	N	5	0	SYSTEM.DBF
ENDTEXT	C	60	0	ENCOUNTR.DBF
ENUMERATE	C	80	0	SF600TEX.DBF
FILE NAME	C	12	0	ENUM.DBF
FINISH	C	30	0	BACKUP.DBF
FIRSTNAME	C	30	0	AREA.DBF
HI	N	9	2	PATIENT.DBF
				VAL.DBF

ID	N	4	0	VAL.DBF
ID	N	5	0	SECTION.DBF
				DISEASE.DBF
				SYMPTOMS.DBF
				SF600TEX.DBF
ID	N	4	0	DICT.DBF
				ENUM.DBF
ID	N	5	0	DIAGNOSE.DBF
				PREMISE.DBF
ID	N	4	0	PATIENT.DBF
LASTNAME	C	30	0	PATIENT.DBF
LIKELY	N	6	2	AREA.DBF
LO	N	9	2	VAL.DBF
MIDDLEINIT	C	1	0	PATIENT.DBF
MUTEX	C	1	0	ENUM.DBF
NAME	C	60	0	DISEASE.DBF
NAME	C	30	0	AREA.DBF
NAME	C	20	0	DICT.DBF
NOPREVSEN	C	60	0	SF600TEX.DBF
NOVICE	L	1	0	SYSTEM.DBF
OBJECT	C	1	0	SECTION.DBF
				SF600TEX.DBF
				ACTION.DBF
				PREMISE.DBF
OP	C	2	0	ACTION.DBF
				PREMISE.DBF
ORD	N	2	0	ENUM.DBF
PASSWORD	C	10	0	SYSTEM.DBF
PATIENT	N	4	0	ENCOUNTR.DBF
PATIENTS	N	4	0	SYSTEM.DBF
PREMISE	N	5	0	RULE.DBF
PRIVACYACT	M	10	0	SYSTEM.DBF
PROBABILTY	N	6	2	DIAGNOSE.DBF
PROBABLE	N	6	2	AREA.DBF
PROMPT	C	80	0	DICT.DBF
PROVIDER	N	4	0	ENCOUNTR.DBF
PROVIDERS	N	4	0	SYSTEM.DBF
QUESTION	C	100	0	DICT.DBF
RATERANK	C	20	0	PATIENT.DBF
REPORT	M	10	0	ENCOUNTR.DBF
REPORT	C	80	0	ENUM.DBF
RULE	N	5	0	RULE.DBF
RULEBASE	C	10	0	AREA.DBF
RULES	N	4	0	AREA.DBF
SALIENCE	N	3	0	RULE.DBF
SECTION	N	3	0	REPORT.DBF
				SECTION.DBF
				SF600TEX.DBF
SEX	C	1	0	PATIENT.DBF
SHOWRULE	L	1	0	SYSTEM.DBF
SIGNON	C	30	0	AREA.DBF
SOCSEC	C	9	0	PATIENT.DBF
SOURCE	C	1	0	DISEASE.DBF
START	C	30	0	AREA.DBF
STARTTEXT	C	60	0	SF600TEX.DBF
STATE	C	1	0	ENCOUNTR.DBF
STATION	C	30	0	SYSTEM.DBF
				ENCOUNTR.DBF
SYMPTOM	N	4	0	SYMPTOMS.DBF
SYSDISCL	M	10	0	SYSTEM.DBF
SYSWARNING	M	10	0	SYSTEM.DBF
TEXT	M	10	0	RULE.DBF

TEXT	M	10	0	RULE.DBF
THRESHOLD	N	6	2	AREA.DBF
TIME	C	8	0	KEYLOG.DBF
				ENCOUNTR.DBF
TREATMENT	M	10	0	ENCOUNTR.DBF
				DISEASE.DBF
UNITS	C	10	0	VAL.DBF
VAL	C	30	0	PREMISE.DBF
				ACTION.DBF
VALUE	C	10	0	SYMPTOMS.DBF
WIDTH	N	2	0	VAL.DBF
XID	N	5	0	DISEASE.DBF

Index Parameter Summary: There are twenty five index associate to the database files; SECTION.IDX, REPORT.IDX, SF600TEX.IDX, AREA.IDX, DICTID.IDX, DICTNAME.IDX, VAL.IDX, ENUM.IDX, DISEASE.IDX, DISID.IDX, PREMISE.IDX, ACTION.IDX, SALIENCE.IDX, RULE.IDX, PATIENT.IDX, PATNAME.IDX, PATID.IDX, ENCOUNTR.IDX, ENCOUNTRD.IDX, ENCOUNTRI.IDX, SYMPTOMS.IDX, DISAREA.IDX, DIAGNOSE.IDX, KEY.IDX, KEYDATE.IDX.

SECTION.IDX -- Indexed on: SECTION

This index file appears to be associated with database(s):
: SECTION.DBF

Used by: SFSYMP() (function in DEMO.PRG)
: REINDEX() (function in U.PRG)

REPORT.IDX -- Indexed on: AREA

This index file appears to be associated with database(s):
: REPORT.DBF

Used by: SFSYMP() (function in DEMO.PRG)

SF600TEX.IDX -- Indexed on: SENTENCE

This index file appears to be associated with database(s):
: SF600TEX.DBF

Used by: SFSYMP() (function in DEMO.PRG)
: SFPRINT (procedure in U.PRG)
: REINDEX() (function in U.PRG)

AREA.IDX -- Indexed on: AREA

This index file appears to be associated with database(s):
: AREA.DBF

Used by: GETEXAREA() (function in U.PRG)
: GETRULEBASE() (function in U.PRG)
: AREANAMES() (function in U.PRG)
: ENCNames() (function in U.PRG)
: SFENCNames() (function in U.PRG)
: OPENFILE() (function in QUESTION.PRG)

DICTID.IDX -- Indexed on: ID

This index file appears to be associated with database(s):
: DICT.DBF

Used by: OPENDICT() (function in U.PRG)

: OPENFILE() (function in QUESTION.PRG)
 DICTNAME.IDX -- Indexed on: NAME
 This index file appears to be associated with database(s):
 : DICT.DBF
 Used by: OPENDICT() (function in U.PRG)
 : REINDEX() (function in U.PRG)
 : OPENFILE() (function in QUESTION.PRG)
 VAL.IDX -- Indexed on: ID
 This index file appears to be associated with database(s):
 : VAL.DBF
 Used by: OPENDICT() (function in U.PRG)
 : REINDEX() (function in U.PRG)
 : OPENFILE() (function in QUESTION.PRG)
 ENUM.IDX -- Indexed on: ID
 This index file appears to be associated with database(s):
 : ENUM.DBF
 Used by: OPENDICT() (function in U.PRG)
 : REINDEX() (function in U.PRG)
 : OPENFILE() (function in QUESTION.PRG)
 DISEASE.IDX -- Indexed on: NAME
 This index file appears to be associated with database(s):
 : DISEASE.DBF
 Used by: DISEASEID() (function in U.PRG)
 : DISNAMES() (function in U.PRG)
 : DISEASEGET() (function in U.PRG)
 : DISEASEPUT() (function in U.PRG)
 : REINDEX() (function in U.PRG)
 DISID.IDX -- Indexed on: ID
 This index file appears to be associated with database(s):
 : DISEASE.DBF
 Used by: DISEASENAM() (function in U.PRG)
 : DISNAMES() (function in U.PRG)
 : REINDEX() (function in U.PRG)
 PREMISE.IDX -- Indexed on: CLAUSE
 This index file appears to be associated with database(s):
 : PREMISE.DBF
 Used by: OPENRULE() (function in U.PRG)
 : REINDEX() (function in U.PRG)
 : OPENFILE() (function in QUESTION.PRG)
 ACTION.IDX -- Indexed on: CLAUSE
 This index file appears to be associated with database(s):
 : ACTION.DBF

Used by: OPENRULE() (function in U.PRG)
: REINDEX() (function in U.PRG)
: OPENFILE() (function in QUESTION.PRG)

SALIENCE.IDX -- Indexed on: SALIENCE

This index file appears to be associated with database(s):
: RULE.DBF

Used by: OPENRULE() (function in U.PRG)
: REINDEX() (function in U.PRG)
: OPENFILE() (function in QUESTION.PRG)

RULE.IDX -- Indexed on: AREA

This index file appears to be associated with database(s):
: RULE.DBF

Used by: OPENRULE() (function in U.PRG)
: REINDEX() (function in U.PRG)
: OPENFILE() (function in QUESTION.PRG)

PATIENT.IDX -- Index on: SOCSEC

This index file appears to be associated with database(s):
: PATIENT.DBF

Used by: REINDEX() (function in U.PRG)

PATNAME.IDX -- Index on: LASTNAME

This index file appears to be associated with database(s):
: PATIENT.DBF

Used by: REINDEX() (function in U.PRG)

PATID.IDX -- Index on: ID

This index file appears to be associated with database(s):
: PATIENT.DBF

Used by: REINDEX() (function in U.PRG)

ENCOUNTR.IDX -- Index on: PATIENT

This index file appears to be associated with database(s):
: ENCOUNTR.DBF

Used by: REINDEX() (function in U.PRG)

ENCOUNTD.IDX -- Index on: DATE

This index file appears to be associated with database(s):
: ENCOUNTR.DBF

Used by: REINDEX() (function in U.PRG)

ENCOUNTI.IDX -- Index on: ENCOUNTRID

This index file appears to be associated with database(s):
: ENCOUNTR.DBF

Used by: REINDEX()
SYMPTOMS.IDX

This index file appears to be associated with database(s):
: SYMPTOMS.DBF

Used by: REINDEX() (function in U.PRG)

DISAREA.IDX -- Index on: AREA

This index file appears to be associated with database(s):
: DISEASE.DBF

Used by: REINDEX() (function in U.PRG)

DIAGNOSE.IDX -- Index on: ID

This index file appears to be associated with database(s):
: DIAGNOSE.DBF

Used by: REINDEX() (function in U.PRG)

KEY.IDX -- Index on: ACTION

This index file appears to be associated with database(s):
: KEYLOG.DBF

Used by: KEYLOG.PRG

KEYDATE.IDX -- Index on: DATE

This index file appears to be associated with database(s):
: KEYLOG.DBF

Used by: KEYLOG.PRG

COLORSET.IDX -- Index on: COLOR_SET

This index file appears to be associated with database(s):
: COLORSET.DBF

Used by: COLORSET.SPR

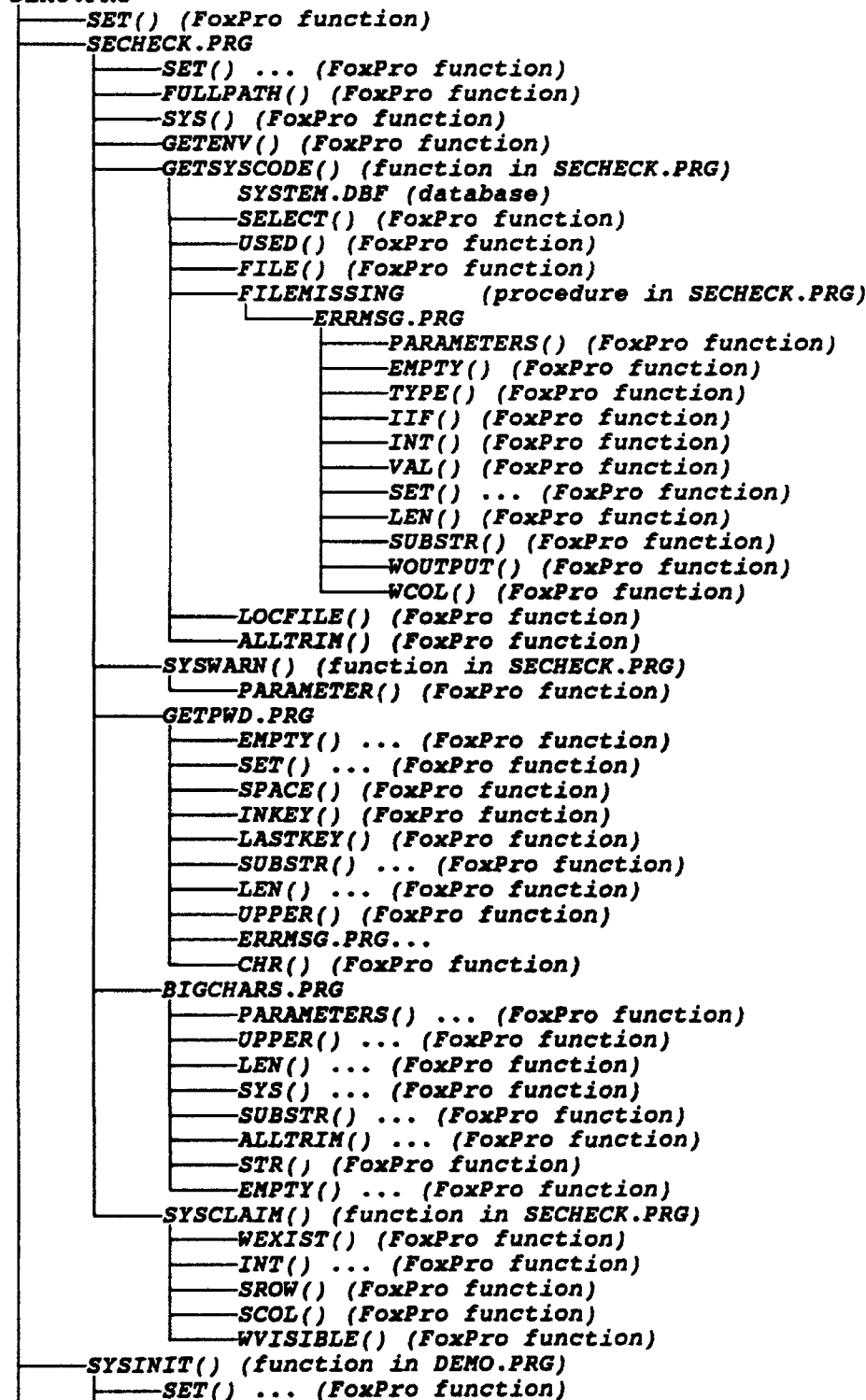
BACKUP.CDX -- Index on: FILE_NAME

This multiple index file appears to be associated with database(s):
: BACKUP.DBF

Used by: PKZIP (procedure in BACKUP.SPR)
: PKUNZIP (procedure in RESTORE.SPR)

Section VI. Tree Diagram. The tree diagram lists each program that is called by the order in which it is used. Under each program there is a list of all functions called.

DEMO.PRG



- EMPTY() ... (FoxPro function)
- GETENV() ... (FoxPro function)
- SETPATH (procedure in U.PRG)
 - EMPTY() ... (FoxPro function)
 - FULLPATH() ... (FoxPro function)
 - IIF() ... (FoxPro function)
 - SET() ... (FoxPro function)
 - CURDIR() (FoxPro function)
- GETOPTS() (function in U.PRG)
 - SYSTEM.DBF (database)
 - SELECT() ... (FoxPro function)
 - ALLTRIM() ... (FoxPro function)
- USERINIT() (function in U.PRG)
 - SET() ... (FoxPro function)
 - INITHELP() (function in U.PRG)
 - OLMR() (function in U.PRG)
 - GETENV() ... (FoxPro function)
 - SYS() ... (FoxPro function)
 - PUSHHELP() (function in U.PRG)
 - ALEN() (FoxPro function)
 - SETHelp() (function in U.PRG)
 - CHR() ... (FoxPro function)
 - GETENV() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - GETCOLORS() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - EOF() (FoxPro function)
 - TRIM() (FoxPro function)
 - PADC() (FoxPro function)
- POPUPSHOW() (function in U.PRG)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)
 - WVISIBLE() ... (FoxPro function)
- INIENCOUNT() (function in DEMO.PRG)
 - CTOD() (FoxPro function)
 - DATE() (FoxPro function)
 - TIME() (FoxPro function)
 - GETSTATION() (function in U.PRG)
 - SYSTEM.DBF (database)
 - SELECT() ... (FoxPro function)
- INISYMPOM() (function in DEMO.PRG)
 - DICTCOUNT() (function in U.PRG)
 - DICT.DBF (database)
 - SELECT() ... (FoxPro function)
 - RECCOUNT() (FoxPro function)
 - DELSYMPOM() (function in DEMO.PRG)
- INIDIAGNOS() (function in DEMO.PRG)
 - DISEASECNT() (function in U.PRG)
 - DISEASE.DBF (database)
 - SELECT() ... (FoxPro function)
 - RECCOUNT() ... (FoxPro function)
 - DELDIAGNOS() (function in DEMO.PRG)
 - RESETDIAG() (function in DEMO.PRG)
- POPUPHIDE() (function in U.PRG)
- IIF() ... (FoxPro function)
- EMPTY() ... (FoxPro function)
- OPENFILES() (function in U.PRG)
 - EMPTY() ... (FoxPro function)
 - POPUPSHOW() ... (function in U.PRG)
 - RIGHT() (FoxPro function)

- OPENPAT() (function in U.PRG)
 - FILEFIND() (function in U.PRG)
 - PARAMETER() ... (FoxPro function)
 - FILE() ... (FoxPro function)
 - DP.PRG
 - PARAMETER() ... (FoxPro function)
 - TYPE() ... (FoxPro function)
 - IIF() ... (FoxPro function)
 - STR() ... (FoxPro function)
 - OCCURS() (FoxPro function)
 - AT() (FoxPro function)
 - LEN() ... (FoxPro function)
 - SUBSTR() ... (FoxPro function)
 - LOCFILE() ... (FoxPro function)
- ERRHAND (procedure in U.PRG)
 - PARAMETER() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - ERRMSG.PRG...
- EMPTY() ... (FoxPro function)
- USED() ... (FoxPro function)
- SELECT() ... (FoxPro function)
- OPENSAMS() (function in U.PRG)
 - EMPTY() ... (FoxPro function)
 - GETENV() ... (FoxPro function)
 - SETPATH... (procedure in U.PRG)
 - FILEFIND() ... (function in U.PRG)
 - USED() ... (FoxPro function)
 - SELECT() ... (FoxPro function)
- OPENENCNT() (function in U.PRG)
 - USED() ... (FoxPro function)
 - SELECT() ... (FoxPro function)
- OPENDICT() (function in U.PRG)
 - DICT.DBF (database)
 - VAL.DBF (database)
 - ENUM.DBF (database)
- OPENRULE() (function in U.PRG)
 - PREMISE.DBF (database)
 - ACTION.DBF (database)
 - RULE.DBF (database)
- POPUPHIDE() ... (function in U.PRG)
- BATCHTEST() (function in DEMO.PRG)
 - RIGHT() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - TIME() ... (FoxPro function)
 - DATE() ... (FoxPro function)
 - EOF() ... (FoxPro function)
 - DELSYMPOM() ... (function in DEMO.PRG)
 - DELDIAGNOS() ... (function in DEMO.PRG)
 - M ENCRULE() (function in DEMO.PRG)
 - LODRULE() (function in DEMO.PRG)
 - GETRULEBASE() (function in U.PRG)
 - AREA.DBF (database)
 - SELECT() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - FOUND() (FoxPro function)
 - TRIM() ... (FoxPro function)
 - ENC2STR() (function in DEMO.PRG)
 - SELECT() ... (FoxPro function)
 - LEN() ... (FoxPro function)
 - SUBSTR() ... (FoxPro function)
 - INVERTCOND() (function in DEMO.PRG)
 - VAL() ... (FoxPro function)

- IIF() ... (FoxPro function)
- ALLTRIM() ... (FoxPro function)
- STR() ... (FoxPro function)
- EVALCOND() (function in DEMO.PRG)
 - MULFACTOR() (function in DEMO.PRG)
 - FACTORID() (function in DEMO.PRG)
 - SET() ... (FoxPro function)
 - ASCAN() (FoxPro function)
 - ERRMESSAGE() (function in U.PRG)
 - AINS() (FoxPro function)
 - IIF() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - SETQ() (function in DEMO.PRG)
 - DEFO() (function in DEMO.PRG)
 - QID() (function in DEMO.PRG)
 - SET() ... (FoxPro function)
 - ASCAN() ... (FoxPro function)
 - ERRMESSAGE() ... (function in U.PRG)
 - AINS() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - LEN() ... (FoxPro function)
 - TRIM() ... (FoxPro function)
 - ASK() (function in DEMO.PRG)
 - EVALQ() (function in DEMO.PRG)
 - QID() ... (function in DEMO.PRG)
 - EMPTY() ... (FoxPro function)
 - EDITSYMP() (function in DEMO.PRG)
 - DICTQ() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - PARAMETERS() ... (FoxPro function)
 - TRIM() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - ERRMESSAGE() ... (function in U.PRG)
 - DICTQV() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - TRIM() ... (FoxPro function)
 - MAX() (FoxPro function)
 - EDNUM() (function in U.PRG)
 - ALEN() ... (FoxPro function)
 - ACOPY() (FoxPro function)
 - PUSHHELP() ... (function in U.PRG)
 - VAL() ... (FoxPro function)
 - W6.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)

- WVISIBLE() ... (FoxPro function)
- LTRIM() (FoxPro function)
- STR() ... (FoxPro function)
- POPHELP() (function in U.PRG)
 - SETHelp() ... (function in U.PRG)
- EDBOOL() (function in U.PRG)
- ALEN() ... (FoxPro function)
- ACOPY() ... (FoxPro function)
- PUSHHELP() ... (function in U.PRG)
- IIF() ... (FoxPro function)
- W6C.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)
 - WVISIBLE() ... (FoxPro function)
- POPHELP() ... (function in U.PRG)
- IIF() ... (FoxPro function)
- LEFT() (FoxPro function)
- DICTECOUNT() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - EOF() ... (FoxPro function)
- DICTQOE() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - EOF() ... (FoxPro function)
 - TRIM() ... (FoxPro function)
- AT() ... (FoxPro function)
- EDCHECK() (function in U.PRG)
 - ALEN() ... (FoxPro function)
 - ACOPY() ... (FoxPro function)
 - PUSHHELP() ... (function in U.PRG)
 - LTRIM() ... (FoxPro function)
 - SETCHECK() (function in U.PRG)
 - ALEN() ... (FoxPro function)
- W6A.PRG
 - SET() ... (FoxPro function)
 - SELECT() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)

- SCOL() ... (FoxPro function)
- WVISIBLE() ... (FoxPro function)
- ALEN() ... (FoxPro function)
- LEN() ... (FoxPro function)
- MIN() (FoxPro function)
- IIF() ... (FoxPro function)
- EDCHECK() (function in U.PRG)
 - VAL() ... (FoxPro function)
 - SUBSTR() ... (FoxPro function)
 - VARREAD() (FoxPro function)
 - AT() ... (FoxPro function)
 - ALEN() ... (FoxPro function)
 - SETCHECK() ... (function in U.PRG)
- IIF() ... (FoxPro function)
- POPHELP() ... (function in U.PRG)
- DEFQ() ... (function in DEMO.PRG)
- PUSHQ() (function in DEMO.PRG)
 - SET() ... (FoxPro function)
 - IIF() ... (FoxPro function)
 - ASCAN() ... (FoxPro function)
- IF() ... (FoxPro function)
- SET() ... (FoxPro function)
- EVALUATE() (FoxPro function)
- LEN() ... (FoxPro function)
- AT() ... (FoxPro function)
- LEFT() ... (FoxPro function)
- SUBSTR() ... (FoxPro function)
- ALLTRIM() ... (FoxPro function)
- STR() ... (FoxPro function)
- ALLTRIM() ... (FoxPro function)
- M_ENCGET() (function in DEMO.PRG)
- POPUPSHOW() ... (function in U.PRG)
- LODENC() (function in DEMO.PRG)
 - ENCGETINFO() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - DELDIAGNOS() ... (function in DEMO.PRG)
 - GETDIAGNOS() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - OPENDIAG() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - ADDDIAGNOS() (function in DEMO.PRG)
 - ALLTRIM() ... (FoxPro function)
 - DEFFACTOR() (function in DEMO.PRG)
 - FACTORID() ... (function in DEMO.PRG)
 - STR() ... (FoxPro function)
 - DELSYMPOM() ... (function in DEMO.PRG)
 - GETSYMPOM() (function in U.PRG)
 - SELECT() ... (FoxPro function)

- OPENSAMP() (function in U.PRG)
 - SELECT() ... (FoxPro function)
- ALLTRIM() ... (FoxPro function)
- TRIM() ... (FoxPro function)
- STR() ... (FoxPro function)
- DICTQ() ... (function in U.PRG)
- DICTENUM() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - EOF() ... (FoxPro function)
 - TRIM() ... (FoxPro function)
- ADDSYMPOM() (function in DEMO.PRG)
 - ALLTRIM() ... (FoxPro function)
 - DEFQ() ... (function in DEMO.PRG)
- M ENCRULE() ... (function in DEMO.PRG)
- AGEAT() (function in DEMO.PRG)
 - YEAR() (FoxPro function)
 - MONTH() (FoxPro function)
 - DAY() (FoxPro function)
 - LTRIM() ... (FoxPro function)
 - STR() ... (FoxPro function)
- EDSEXAGE() (function in DEMO.PRG)
 - ADDSYMPOM() (function in DEMO.PRG)
 - ALLEN() ... (FoxPro function)
 - DICTQNAME() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - ALLTRIM() ... (FoxPro function)
 - STR() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - ERASEQ() (function in DEMO.PRG)
 - QID() ... (function in DEMO.PRG)
 - ADDSYMPOM() ... (function in DEMO.PRG)
- POPUPHIDE() ... (function in U.PRG)
- M ASSIST (procedure in DEMO.PRG)
 - EMPTY() ... (FoxPro function)
 - POPUPSHOW() ... (function in U.PRG)
 - RESETDIAG() ... (function in DEMO.PRG)
 - TESTRULES() (function in DEMO.PRG)
 - SELECT() ... (FoxPro function)
 - EOF() ... (FoxPro function)
 - EVALCOND() ... (function in DEMO.PRG)
 - ALLTRIM() ... (FoxPro function)
 - STR() ... (FoxPro function)
 - EVALRULES() (function in DEMO.PRG)
 - SELECT() ... (FoxPro function)
 - ALLEN() ... (FoxPro function)
 - EVALCOND() ... (function in DEMO.PRG)
 - ALLTRIM() ... (FoxPro function)
 - STR() ... (FoxPro function)
 - NAKEDIAG() (function in DEMO.PRG)
 - NORMFACTOR() (function in DEMO.PRG)
 - SUMFACTOR() (function in DEMO.PRG)
 - POPMESSAGE() (function in U.PRG)
 - PUSHHELP() ... (function in U.PRG)
 - W9.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)

- WVISIBLE() ... (FoxPro function)
 - POPHELP() ... (function in U.PRG)
 - POPUPHIDE() ... (function in U.PRG)
 - RELEASE() (function in U.PRG)
 - PARAMETER() ... (FoxPro function)
 - WONTOP() (FoxPro function)
 - OBJNUM() (FoxPro function)
- M_ENCSAV() (function in DEMO.PRG)
 - POPMESSAGE() ... (function in U.PRG)
- M_ENCVALID() (function in DEMO.PRG)
 - EMPTY() ... (FoxPro function)
- ADDENCNT() (function in U.PRG)
 - NEXTENCNT() (function in U.PRG)
 - SYSTEM.DBF (database)
 - SELECT() ... (FoxPro function)
 - IIF() ... (FoxPro function)
 - TYPE() ... (FoxPro function)
- NEWENCNT() (function in U.PRG)
 - SELECT() ... (FoxPro function)
- PUTENCNT() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
- PUTSYMPTOM() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - OPENSYP() ... (function in U.PRG)
 - FOUND() ... (FoxPro function)
 - ALLEN() ... (FoxPro function)
 - IIF() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - SPLIT() (function in U.PRG)
 - AT() ... (FoxPro function)
 - LEFT() ... (FoxPro function)
 - SUBSTR() ... (FoxPro function)
 - LEN() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - DICTQ() ... (function in U.PRG)
 - DICTORD() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - EOF() ... (FoxPro function)
 - ALLTRIM() ... (FoxPro function)
 - STR() ... (FoxPro function)
 - EOF() ... (FoxPro function)
- PUTDIAGNOS() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - OPENDIAG() ... (function in U.PRG)
 - FOUND() ... (FoxPro function)
 - ALLEN() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - VAL() ... (FoxPro function)
 - EOF() ... (FoxPro function)
- STR() ... (FoxPro function)
- ENC2BIT() (function in DEMO.PRG)
 - ALLEN() ... (FoxPro function)
 - IIF() ... (FoxPro function)
- ALLEN() ... (FoxPro function)
- DISEASENAM() (function in U.PRG)
 - DISEASE.DBF (database)
 - SELECT() ... (FoxPro function)
 - PARAMETERS() ... (FoxPro function)
 - ALLEN() ... (FoxPro function)

- IIF() ... (FoxPro function)
- FOUND() ... (FoxPro function)
- TRIM() ... (FoxPro function)
- VAL() ... (FoxPro function)
- IIF() ... (FoxPro function)
- LEFT() ... (FoxPro function)
- INKEY() ... (FoxPro function)
- MAINMENU() (function in U.PRG)
- CAMD.MPR
 - EMPTY() ... (FoxPro function)
 - WONTOP() ... (FoxPro function)
 - LOCFILE() ... (FoxPro function)
 - Q9POPFL0Y (procedure in CAMD.MPR)
 - PUSHHELP() ... (function in U.PRG)
 - PWDEDT.PRG
 - SYSTEM.DBF (database)
 - SET() ... (FoxPro function)
 - FULLPATH() ... (FoxPro function)
 - SYS() ... (FoxPro function)
 - GETENV() ... (FoxPro function)
 - SPACE() ... (FoxPro function)
 - INKEY() ... (FoxPro function)
 - LASTKEY() ... (FoxPro function)
 - SUBSTR() ... (FoxPro function)
 - LEN() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - UPPER() ... (FoxPro function)
 - CHR() ... (FoxPro function)
 - POPHELP() ... (function in U.PRG)
 - MHIT (procedure in U.PRG)
 - RDLEVEL() (FoxPro function)
 - REINDEX() (function in U.PRG)
 - PATIENT.DBF (database)
 - ENCOUNTR.DBF (database)
 - SYMPTOMS.DBF (database)
 - DICT.DBF (database)
 - VAL.DBF (database)
 - ENUM.DBF (database)
 - PREMISE.DBF (database)
 - ACTION.DBF (database)
 - RULE.DBF (database)
 - DISEASE.DBF (database)
 - DIAGNOSE.DBF (database)
 - REPORT.DBF (database)
 - SECTION.DBF (database)
 - SF600TEX.DBF (database)
 - EMPTY() ... (FoxPro function)
 - WOUTPUT() ... (FoxPro function)
 - POPUPSHOW() ... (function in U.PRG)
 - CLOSEFILES() (function in U.PRG)
 - POPUPHIDE() ... (function in U.PRG)
 - QUIT (procedure in U.PRG)
 - EMPTY() ... (FoxPro function)
 - WOUTPUT() ... (FoxPro function)
 - POPMESSAGE() ... (function in U.PRG)
 - Q9POPFLR8 (procedure in CAMD.MPR)
 - CAMDTR.MPR
 - EMPTY() ... (FoxPro function)
 - WONTOP() ... (FoxPro function)
 - PWDEDT.PRG...
 - MHIT... (procedure in U.PRG)
 - REINDEX() ... (function in U.PRG)

```

QUIT...      (procedure in U.PRG)
MHIT...      (procedure in U.PRG)
CAND.MPR...
COLORSET.SPR
  COLORSET.DBF (database)
  SET() ... (FoxPro function)
  WEXIST() ... (FoxPro function)
  INT() ... (FoxPro function)
  SROW() ... (FoxPro function)
  SCOL() ... (FoxPro function)
  USED() ... (FoxPro function)
  WVISIBLE() ... (FoxPro function)
  Q9POPF2H() (function in COLORSET.SPR)
  TRIM() ... (FoxPro function)
MYHANDLER() (function in U.PRG)
LEN() ... (FoxPro function)
WONTOP() ... (FoxPro function)
M_RESUME     (procedure in DEMO.PRG)
  ALEN() ... (FoxPro function)
  STRCOND() (function in DEMO.PRG)
  PARAMETERS() ... (FoxPro function)
  TYPE() ... (FoxPro function)
  LEN() ... (FoxPro function)
  LOOKUPNAM() (function in DEMO.PRG)
  DICTQ() ... (function in U.PRG)
  EMPTY() ... (FoxPro function)
  LEFT() ... (FoxPro function)
  STRDIAG() (function in DEMO.PRG)
  AINS() ... (FoxPro function)
  IIF() ... (FoxPro function)
  TYPE() ... (FoxPro function)
  DISEASENAM() ... (function in U.PRG)
  VAL() ... (FoxPro function)
  PARAMETERS() ... (FoxPro function)
  LTRIM() ... (FoxPro function)
  STR() ... (FoxPro function)
  GETEXAREA() (function in U.PRG)
  AREA.DBF (database)
  EMPTY() ... (FoxPro function)
  SELECT() ... (FoxPro function)
  IIF() ... (FoxPro function)
  FOUND() ... (FoxPro function)
  TRIM() ... (FoxPro function)
  DISEASENAM() ... (function in U.PRG)
  AGEAT() ... (function in DEMO.PRG)
  EDSEXAGE() ... (function in DEMO.PRG)
  ENCNTTITLE() (function in U.PRG)
  PUSHHELP() ... (function in U.PRG)
W12.SPR
  SET() ... (FoxPro function)
  WEXIST() ... (FoxPro function)
  WVISIBLE() ... (FoxPro function)
  EDITCOMPL() (function in U.PRG)
  M_COMPL    (procedure in DEMO.PRG)
  EMPTY() ... (FoxPro function)
  GETDIAGAREA() (function in DEMO.PRG)
  AREACOUNT() (function in U.PRG)
  AREA.DBF (database)
  SELECT() ... (FoxPro function)

```

- RECCOUNT() ... (FoxPro function)
- AREANAMES() (function in U.PRG)
 - AREA.DBF (database)
 - SELECT() ... (FoxPro function)
 - EOF() ... (FoxPro function)
 - TRIM() ... (FoxPro function)
- GETAREALIST() (function in U.PRG)
 - ALLEN() ... (FoxPro function)
 - PUSHHELP() ... (function in U.PRG)
 - W5.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)
 - WVISIBLE() ... (FoxPro function)
 - POPHELP() ... (function in U.PRG)
- IIF() ... (FoxPro function)
- M_ENCRULE() ... (function in DEMO.PRG)
- POPMESSAGE() ... (function in U.PRG)
- EMPTY() ... (FoxPro function)
- GETEXAREA() ... (function in U.PRG)
- OBJNUM() ... (FoxPro function)
- EDITHANDP() (function in U.PRG)
- M HANDP (procedure in DEMO.PRG)
 - EMPTY() ... (FoxPro function)
 - POPUPSHOW() ... (function in U.PRG)
 - TESTRULES() ... (function in DEMO.PRG)
 - POPUPHIDE() ... (function in U.PRG)
 - POPMESSAGE() ... (function in U.PRG)
 - RELEASE() ... (function in U.PRG)
 - OBJNUM() ... (FoxPro function)
- STRCOND() ... (function in DEMO.PRG)
- EDITCOND() (function in U.PRG)
- M REDOQ() (function in DEMO.PRG)
 - GETQ() (function in DEMO.PRG)
 - QID() ... (function in DEMO.PRG)
 - IIF() ... (FoxPro function)
 - ERASEQ() ... (function in DEMO.PRG)
 - ASK() ... (function in DEMO.PRG)
 - RESTOREQ() (function in DEMO.PRG)

- QID() ... (function in DEMO.PRG)
- STRCOND() ... (function in DEMO.PRG)
- EDITDIAG() (function in U.PRG)
- M_DIAG (procedure in DEMO.PRG)
 - EMPTY() ... (FoxPro function)
 - ASKDIAG() (function in DEMO.PRG)
 - DISEASECNT() ... (function in U.PRG)
 - DISNAMES() (function in U.PRG)
 - DISEASE.DBF (database)
 - SELECT() ... (FoxPro function)
 - PARAMETERS() ... (FoxPro function)
 - EOF() ... (FoxPro function)
 - TRIM() ... (FoxPro function)
 - IIF() ... (FoxPro function)
 - GETDISLIST() (function in U.PRG)
 - ALN() ... (FoxPro function)
 - PUSHHELP() ... (function in U.PRG)
 - W7.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)
 - WVISIBLE() ... (FoxPro function)
 - POPHELP() ... (function in U.PRG)
 - DISEASEGET() (function in U.PRG)
 - DISEASE.DBF (database)
 - SELECT() ... (FoxPro function)
 - UPPER() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - CHR() ... (FoxPro function)
 - W8.PRG
 - SET() ... (FoxPro function)
 - SELECT() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)

- INT() ... (FoxPro function)
- SROW() ... (FoxPro function)
- SCOL() ... (FoxPro function)
- WVISIBLE() ... (FoxPro function)
- DISEASENAM() ... (function in U.PRG)
- DISEASEID() (function in U.PRG)
 - DISEASE.DBF (database)
 - SELECT() ... (FoxPro function)
 - PARAMETERS() ... (FoxPro function)
 - ALen() ... (FoxPro function)
 - IIF() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
- POPMESSAGE() ... (function in U.PRG)
- OBJNUM() ... (FoxPro function)
- EMPTY() ... (FoxPro function)
- DISEASENAM() ... (function in U.PRG)
- OBJNUM() ... (FoxPro function)
- EDITASSIS() (function in U.PRG)
 - M ASSIST... (procedure in DEMO.PRG)
 - STRDIAG() ... (function in DEMO.PRG)
- EDITVDIAG() (function in U.PRG)
 - DISEASEGET() ... (function in U.PRG)
 - PUSHHELP() ... (function in U.PRG)
 - W8.PRG...
 - POPHELP() ... (function in U.PRG)
- EDITTREAT() (function in U.PRG)
 - M TREAT (procedure in DEMO.PRG)
 - EMPTY() ... (FoxPro function)
 - OBJNUM() ... (FoxPro function)
 - POPMESSAGE() ... (function in U.PRG)
 - RELEASE() ... (function in U.PRG)
- Q9VOUNCRG() (function in W12.SPR)
 - WONTOP() ... (FoxPro function)
 - W12C.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)
 - WVISIBLE() ... (FoxPro function)
 - Q740N41W1() (function in W12C.SPR)
 - WONTOP() ... (FoxPro function)
 - W12C.SPR...
 - Q740N423H() (function in W12C.SPR)
 - RELEASE() ... (function in U.PRG)
- CHR() ... (FoxPro function)
- CHR() ... (FoxPro function)
- EVALID() (function in U.PRG)
 - PUSHHELP() ... (function in U.PRG)
 - W2.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)

```

INT() ... (FoxPro function)
SROW() ... (FoxPro function)
SCOL() ... (FoxPro function)
WVISIBLE() ... (FoxPro function)
Q7IOLH8OT() (function in W2.SPR)
Q7IOLH8WD() (function in W2.SPR)
  PRINTENC() (function in
    U.PRG)
    ALLTRIM() ... (FoxPro
      function)
    EMPTY() ... (FoxPro
      function)
    DTOC() (FoxPro function)
    STR() ... (FoxPro
      function)
    TRIM() ... (FoxPro
      function)
    IIF() ... (FoxPro
      function)
    ALLEN() ... (FoxPro
      function)
    PROPER() (FoxPro
      function)
    ALLTRIM() ... (FoxPro
      function)
  Q7IOLH94Q() (function in W2.SPR)
  OBJNUM() ... (FoxPro function)
PCPHELP() ... (function in U.PRG)
Q9VOUMD2M() (function in W12.SPR)
RELEASE() ... (function in U.PRG)
Q9VOUMDDI() (function in W12.SPR)
RELEASE() ... (function in U.PRG)
Q9VOUMDJJ() (function in W12.SPR)
WONTOP() ... (FoxPro function)
W12A.SPR
  SET() ... (FoxPro function)
  WEXIST() ... (FoxPro function)
  INT() ... (FoxPro function)
  SROW() ... (FoxPro function)
  SCOL() ... (FoxPro function)
  WVISIBLE() ... (FoxPro function)
  EDITCOND() ... (function in U.PRG)
  EDITHANDP() ... (function in
    U.PRG)
  Q740N3U9Y() (function in W12A.SPR)
  RELEASE() ... (function in
    U.PRG)
  CHR() ... (FoxPro function)
  Q740N3UHF() (function in W12A.SPR)
  WONTOP() ... (FoxPro function)
  W12A.SPR...
Q9VOUMDQF() (function in W12.SPR)
RELEASE() ... (function in U.PRG)
Q9VOUMDWG() (function in W12.SPR)
WONTOP() ... (FoxPro function)
W12B.SPR
  SET() ... (FoxPro function)
  WEXIST() ... (FoxPro function)
  INT() ... (FoxPro function)
  SROW() ... (FoxPro function)
  SCOL() ... (FoxPro function)
  WVISIBLE() ... (FoxPro function)

```

- Q740N3XWA() (function in W12B.SPR)
 - RELEASE() ... (function in U.PRG)
 - EDITVDIAG() ... (function in U.PRG)
 - EDITASSIS() ... (function in U.PRG)
 - Q740N3Y5O() (function in W12B.SPR)
 - WONTOP() ... (FoxPro function)
 - W12B.SPR...
 - CHR() ... (FoxPro function)
 - Q9VOLUME80() (function in W12.SPR)
 - WONTOP() ... (FoxPro function)
 - STOPREAD() (function in U.PRG)
 - WVISIBLE() ... (FoxPro function)
 - WREAD() (FoxPro function)
 - POPHELP() ... (function in U.PRG)
 - M_ENCSAV() ... (function in DEMO.PRG)
 - M_ENCCLO() (function in DEMO.PRG)
 - M_ENCVALID() ... (function in DEMO.PRG)
 - WCONFIRM() (function in U.PRG)
 - PUSHHELP() ... (function in U.PRG)
 - W10.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)
 - WVISIBLE() ... (FoxPro function)
 - POPHELP() ... (function in U.PRG)
 - M_REGPAT (procedure in DEMO.PRG)
 - CTOD() ... (FoxPro function)
 - GETCURPAT() (function in DEMO.PRG)
 - PATID() (function in U.PRG)
 - PUSHHELP() ... (function in U.PRG)
 - W4.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)
 - WVISIBLE() ... (FoxPro function)
 - PVALID() (function in U.PRG)
 - Q7Q0ZD5UR() (function in W4.SPR)
 - Q7Q0ZD65J() (function in W4.SPR)
 - M_PATLIST (procedure in DEMO.PRG)
 - GETPATIENT() (function in DEMO.PRG)
 - PATCOUNT() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - RECCOUNT() ... (FoxPro function)
 - PATNAMES() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - SET() ... (FoxPro function)
 - TYPE() ... (FoxPro function)
 - ALLTRIM() ... (FoxPro function)

- ALLEN() ... (FoxPro function)
- EOF() ... (FoxPro function)
- STRNAME() (function in U.PRG)
 - IIF() ... (FoxPro function)
 - EMPTY() ... (FoxPro function)
 - TRIM() ... (FoxPro function)
- IIF() ... (FoxPro function)
- GETPATLIST() (function in U.PRG)
 - ALLEN() ... (FoxPro function)
 - IIF() ... (FoxPro function)
 - TYPE() ... (FoxPro function)
 - ALLTRIM() ... (FoxPro function)
 - PUSHHELP() ... (function in U.PRG)
 - W3.SPR
 - SET() ... (FoxPro function)
 - WEXIST() ... (FoxPro function)
 - INT() ... (FoxPro function)
 - SROW() ... (FoxPro function)
 - SCOL() ... (FoxPro function)
 - WVISIBLE() ... (FoxPro function)
 - POPHELP() ... (function in U.PRG)
- SUBSTR() ... (FoxPro function)
- AT() ... (FoxPro function)
- EMPTY() ... (FoxPro function)
- M_PATGET() (function in DEMO.PRG)
 - PATGETINFO() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - FOUND() ... (FoxPro function)
 - ERRMESSAGE() ... (function in U.PRG)
- STOPREAD() ... (function in U.PRG)
- POPHELP() ... (function in U.PRG)
- M_PATPUT() (function in DEMO.PRG)
 - IIF() ... (FoxPro function)
 - TYPE() ... (FoxPro function)
 - POPMESSAGE() ... (function in U.PRG)
 - ISVALIDSSN() (function in U.PRG)
 - SELECT() ... (FoxPro function)
 - FOUND() ... (FoxPro function)

```

PATPUTINFO() (function in U.PRG)
  └── SELECT() ... (FoxPro function)
      └── FOUND() ... (FoxPro function)
M_PATADD() (function in DEMO.PRG)
  ├── IIF() ... (FoxPro function)
  ├── TYPE() ... (FoxPro function)
  ├── POPMESSAGE() ... (function in U.PRG)
  ├── ISVALIDSSN() ... (function in U.PRG)
  ├── ADDPAT() (function in U.PRG)
  │   └── NEXTPAT() (function in U.PRG)
  │       ├── SYSTEM.DBF (database)
  │       ├── SELECT() ... (FoxPro function)
  │       ├── IIF() ... (FoxPro function)
  │       └── TYPE() ... (FoxPro function)
  └── NEWPATIENT() (function in U.PRG)
      └── SELECT() ... (FoxPro function)
M_ENCVALID() ... (function in DEMO.PRG)
SETCURPAT() (function in DEMO.PRG)
AGEAT() ... (function in DEMO.PRG)
USERCLOSE() (function in U.PRG)
  └── EMPTY() ... (FoxPro function)
SYSCLOSE() (function in DEMO.PRG)

```

Section VII. Procedure and Function Summary. There are twenty one (21) files containing procedures in the CAMD system: DEMO.PRG, U.PRG, W12.SPR, W2.SPR, WP1.PRG, WP2.PRG, WPATS.PRG, WP3.SPR, WP7.SPR, SECHECK.PRG, W12A.SPR, W12B.SPR, W12C.SPR, W4.SPR, BACKUP.SPR, RESTORE.SPR, W7A.SPR, QUESTION.PRG, Q2.PRG, COLORSET.SPR, CAMD.MPR.

DEMO.PRG -- Last updated: 10/05/92 at 16:36

Contains: M_RESUME	(Params: none)
Called by: M_ENC	(procedure in DEMO.PRG)
Called by: MYHANDLER	(procedure in U.PRG)
Calls: STRCOND	(procedure in DEMO.PRG)
Calls: STRDIAG	(procedure in DEMO.PRG)
Calls: GETEXAREA()	(function in U.PRG)
Calls: DISEASENAM()	(function in U.PRG)
Calls: AGEAT	(procedure in DEMO.PRG)
Calls: EDSEXAGE	(procedure in DEMO.PRG)
Calls: ENCNTTITLE()	(function in U.PRG)
Calls: M_ENCSAV	(procedure in DEMO.PRG)
Calls: M_ENCCLO	(procedure in DEMO.PRG)
Calls: M_ENCVVALID	(procedure in DEMO.PRG)
Calls: WCONFIRM()	(function in U.PRG)
Contains: M_REGPAT	(Params: none)
Called by: MYHANDLER	(procedure in U.PRG)
Calls: GETCURPAT	(procedure in DEMO.PRG)
Calls: PATID()	(function in U.PRG)
Calls: M_PATPUT	(procedure in DEMO.PRG)
Calls: M_PATADD	(procedure in DEMO.PRG)
Calls: M_ENCVVALID	(procedure in DEMO.PRG)
Calls: SETCURPAT	(procedure in DEMO.PRG)
Calls: AGEAT	(procedure in DEMO.PRG)
Contains: M_PAT	(Params: none)
Calls: OPENFILES()	(function in U.PRG)
Calls: KEYLOG.PRG	
Calls: WPAT	(procedure in DEMO.PRG)
Calls: FLUSHFILES()	(function in U.PRG)
Contains: M_ENC	(Params: none)
Called by: M_NEWENC	(procedure in DEMO.PRG)
Called by: M_TRAINING	(procedure in DEMO.PRG)
Calls: ENCOUNT	(procedure in DEMO.PRG)
Calls: M_RESUME	(procedure in DEMO.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Contains: M_COMPL	(Params: none)
Called by: EDITCOMPL	(procedure in U.PRG)
Calls: GETDIAGAREA	(procedure in DEMO.PRG)
Calls: M_ENCRULE	(procedure in DEMO.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Contains: M_DIAG	(Params: none)
Called by: EDITDIAG	(procedure in U.PRG)
Calls: ASKDIAG	(procedure in DEMO.PRG)
Calls: DISEASENAM()	(function in U.PRG)
Calls: DISEASEID()	(function in U.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Contains: M_ASSIST	(Params: none)
Called by: BATCHTEST()	(function in DEMO.PRG)
Called by: EDITASSIS	(procedure in U.PRG)
Calls: POPUPSHOW()	(function in U.PRG)
Calls: RESETDIAG	(procedure in DEMO.PRG)
Calls: TESTRULES	(procedure in DEMO.PRG)
Calls: EVALRULES	(procedure in DEMO.PRG)
Calls: MAKEDIAG	(procedure in DEMO.PRG)

Calls: POPMESSAGE()	(function in U.PRG)
Calls: POPUPHIDE()	(function in U.PRG)
Contains: M_HANDP	(Params: none)
Called by: EDITHANDP	(procedure in U.PRG)
Calls: POPUPSHOW()	(function in U.PRG)
Calls: TESTRULES	(procedure in DEMO.PRG)
Calls: POPUPHIDE()	(function in U.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Contains: M_TREAT	(Params: NEWTREAT)
Called by: EDITTREAT	(procedure in U.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Contains: M_REDOQ	(Params: MITEM)
Called by: EDITCOND	(procedure in U.PRG)
Calls: GETQ	(procedure in DEMO.PRG)
Calls: ERASEQ	(procedure in DEMO.PRG)
Calls: ASK	(procedure in DEMO.PRG)
Calls: RESTOREQ	(procedure in DEMO.PRG)
Contains: ENCOUNT	(Params: none)
Called by: M_ENC	(procedure in DEMO.PRG)
Calls: GETENCNT	(procedure in DEMO.PRG)
Calls: PATSSN()	(function in U.PRG)
Calls: M_PATGET	(procedure in DEMO.PRG)
Calls: M_ENCGET	(procedure in DEMO.PRG)
Calls: M_ENCNEW	(procedure in DEMO.PRG)
Contains: M_ENCCLO	(Params: none)
Called by: M_RESUME	(procedure in DEMO.PRG)
Contains: M_ENCSAV	(Params: none)
Called by: BATCHTEST()	(function in DEMO.PRG)
Called by: M_RESUME	(procedure in DEMO.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Calls: M_ENCVALID	(procedure in DEMO.PRG)
Calls: ADDENCNT()	(function in U.PRG)
Calls: NEWENCNT()	(function in U.PRG)
Calls: PUTENCNT()	(function in U.PRG)
Calls: PUTSYMPTOM()	(function in U.PRG)
Calls: PUTDIAGNOS()	(function in U.PRG)
Contains: M_ENCVALID	(Params: none)
Called by: M_RESUME	(procedure in DEMO.PRG)
Called by: M_REGPAT	(procedure in DEMO.PRG)
Called by: M_ENCSAV	(procedure in DEMO.PRG)
Contains: M_ENCGET	(Params: none)
Called by: BATCHTEST()	(function in DEMO.PRG)
Called by: ENCOUNT	(procedure in DEMO.PRG)
Calls: POPUPSHOW()	(function in U.PRG)
Calls: LODENC	(procedure in DEMO.PRG)
Calls: M_ENCRULE	(procedure in DEMO.PRG)
Calls: AGEAT	(procedure in DEMO.PRG)
Calls: EDSEXAGE	(procedure in DEMO.PRG)
Calls: POPUPHIDE()	(function in U.PRG)
Contains: M_ENCNEW	(Params: none)
Called by: ENCOUNT	(procedure in DEMO.PRG)
Calls: NEXTENCNT()	(function in U.PRG)
Calls: AGEAT	(procedure in DEMO.PRG)
Calls: DELSYMPTOM	(procedure in DEMO.PRG)
Calls: DELDIAGNOS	(procedure in DEMO.PRG)
Calls: EDSEXAGE	(procedure in DEMO.PRG)
Contains: EDSEXAGE	(Params: MAGE, MSEX)
Called by: M_RESUME	(procedure in DEMO.PRG)
Called by: M_ENCGET	(procedure in DEMO.PRG)
Called by: M_ENCNEW	(procedure in DEMO.PRG)
Calls: ADDHISTORY	(procedure in DEMO.PRG)
Contains: M_ENCRULE	(Params: none)

Called by: BATCHTEST()	(function in DEMO.PRG)
Called by: M_COMPL	(procedure in DEMO.PRG)
Called by: M_ENCGET	(procedure in DEMO.PRG)
Calls: LODRULE	(procedure in DEMO.PRG)
Contains: LODRULE	(Params: MAREA)
Called by: M_ENCRULE	(procedure in DEMO.PRG)
Calls: GETRULEBASE()	(function in U.PRG)
Contains: TESTRULES	(Params: MAREA)
Called by: M_ASSIST	(procedure in DEMO.PRG)
Called by: M_HANDP	(procedure in DEMO.PRG)
Calls: EVALCOND	(procedure in DEMO.PRG)
Contains: EVALRULES	(Params: MAREA)
Called by: M_ASSIST	(procedure in DEMO.PRG)
Calls: EVALCOND	(procedure in DEMO.PRG)
Contains: OBJTYPE	(Params: MOBJECT)
Contains: EVALCOND	(Params: MOBJECT, MID, MOP, MVAL)
Called by: TESTRULES	(procedure in DEMO.PRG)
Called by: EVALRULES	(procedure in DEMO.PRG)
Called by: INVERTCOND	(procedure in DEMO.PRG)
Calls: MULFACTOR	(procedure in DEMO.PRG)
Calls: SETQ	(procedure in DEMO.PRG)
Calls: ASK	(procedure in DEMO.PRG)
Contains: UNDO RULES	(Params: none)
Called by: M_UNDO RULE	(procedure in DEMO.PRG)
Calls: UNDOFACTOR	(procedure in DEMO.PRG)
Calls: UNDOQUAL	(procedure in DEMO.PRG)
Contains: UNDOFACTOR	(Params: STATESTR)
Called by: UNDO RULES	(procedure in DEMO.PRG)
Calls: FACTORID	(procedure in DEMO.PRG)
Contains: UNDOQUAL	(Params: STATESTR)
Called by: UNDO RULES	(procedure in DEMO.PRG)
Called by: POPQ	(procedure in DEMO.PRG)
Calls: QID	(procedure in DEMO.PRG)
Contains: ASKDIAG	(Params: MDIAGNOSIS)
Called by: M_DIAG	(procedure in DEMO.PRG)
Calls: DISEASECNT()	(function in U.PRG)
Calls: DISNAMES()	(function in U.PRG)
Calls: GETDISLIST()	(function in U.PRG)
Contains: GETDIAGAREA	(Params: none)
Called by: M_COMPL	(procedure in DEMO.PRG)
Called by: M_DATA	(procedure in DEMO.PRG)
Calls: AREACOUNT()	(function in U.PRG)
Calls: AREANAMES()	(function in U.PRG)
Calls: GETAREALIST()	(function in U.PRG)
Contains: GETPATIENT	(Params: MKEY)
Called by: M_PATLIST	(procedure in DEMO.PRG)
Calls: PATCOUNT()	(function in U.PRG)
Calls: PATNAMES()	(function in U.PRG)
Calls: GETPATLIST()	(function in U.PRG)
Contains: GETENCNT	(Params: MPATIENTID, MDATE)
Called by: ENCOUNT	(procedure in DEMO.PRG)
Calls: ENCCOUNT()	(function in U.PRG)
Calls: ENCNames()	(function in U.PRG)
Calls: GETENCLIST()	(function in U.PRG)
Calls: PATSSN()	(function in U.PRG)
Contains: INIENCOUNT()	(Params: none)
Called by: DEMO.PRG	
Calls: GETSTATION()	(function in U.PRG)
Contains: LODENC	(Params: MENCOUNTID)
called by: M_ENCGET	(procedure in DEMO.PRG)
Called by: SFENCOUNT	(procedure in U.PRG)
Calls: ENCGETINFO()	(function in U.PRG)

Calls: DELDIAGNOS	(procedure in DEMO.PRG)
Calls: GETDIAGNOS()	(function in U.PRG)
Calls: DELSYMPTON	(procedure in DEMO.PRG)
Calls: GETSYMPTON()	(function in U.PRG)
Contains: M_CHANGE	(Params: none)
Calls: CHANGEQ	(procedure in DEMO.PRG)
Calls: POPO	(procedure in DEMO.PRG)
Calls: M_UNDORULE	(procedure in DEMO.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Contains: M_UNDO	(Params: none)
Calls: LASTQ	(procedure in DEMO.PRG)
Calls: POPO	(procedure in DEMO.PRG)
Calls: M_UNDORULE	(procedure in DEMO.PRG)
Contains: M_UNDORULE	(Params: none)
Called by: M_CHANGE	(procedure in DEMO.PRG)
Called by: M_UNDO	(procedure in DEMO.PRG)
Calls: UNDO RULES	(procedure in DEMO.PRG)
Contains: MYUNDO	(Params: none)
Calls: UNDOMENU()	(function in U.PRG)
Contains: MAKEDIAG	(Params: none)
Called by: M_ASSIST	(procedure in DEMO.PRG)
Calls: NORMFACTOR	(procedure in DEMO.PRG)
Contains: GETCURPAT	(Params: MPID, MSSN, MLAST, MFIRST, MBIRTH, MSX, MRATE)
Called by: M_REGPAT	(procedure in DEMO.PRG)
Contains: SETCURPAT	(Params: MPID, MSSN, MLAST, MFIRST, MBIRTH, MSX, MRATE)
Called by: M_REGPAT	(procedure in DEMO.PRG)
Contains: M_PATLIST	(Params: none)
Called by: Q7Q0ZD65J()	(function in W4.SPR)
Calls: GETPATIENT	(procedure in DEMO.PRG)
Calls: M_PATGET	(procedure in DEMO.PRG)
Contains: M_PATGET	(Params: MPID, MSSN, MLAST, MFIRST, MBIRTH, MSX, MRATE)
Called by: ENCOUNT	(procedure in DEMO.PRG)
Called by: M_PATLIST	(procedure in DEMO.PRG)
Called by: SFENCOUNT	(procedure in U.PRG)
Calls: PATGETINFO()	(function in U.PRG)
Calls: ERRMESSAGE()	(function in U.PRG)
Contains: M_PATPUT	(Params: MPID, MSSN, MLAST, MFIRST, MBIRTH, MSX, MRATE)
Called by: M_REGPAT	(procedure in DEMO.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Calls: ISVALIDSSN()	(function in U.PRG)
Calls: PATPUTINFO()	(function in U.PRG)
Contains: M_PATADD	(Params: MPID, MSSN, MLAST, MFIRST, MBIRTH, MSX, MRATE)
Called by: M_REGPAT	(procedure in DEMO.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Calls: ISVALIDSSN()	(function in U.PRG)
Calls: ADDPAT()	(function in U.PRG)
Calls: NEWPATIENT()	(function in U.PRG)
Contains: AGEAT	(Params: MDATE, MDATEBIRTH)
Called by: M_RESUME	(procedure in DEMO.PRG)
Called by: M_REGPAT	(procedure in DEMO.PRG)
Called by: M_ENCGET	(procedure in DEMO.PRG)
Called by: M_ENCNEW	(procedure in DEMO.PRG)
Called by: WPAT	(procedure in DEMO.PRG)
Contains: ADDDIAGNOS	(Params: DNAME, DVAL)
Called by: GETDIAGNOS()	(function in U.PRG)
Calls: DEFFACTOR	(procedure in DEMO.PRG)
Contains: ADDHISTORY	(Params: HISTNAME, HISTVAL)

Called by: EDSEXAGE	(procedure in DEMO.PRG)
Calls: DICTQNAME()	(function in U.PRG)
Calls: ERASEQ	(procedure in DEMO.PRG)
Calls: ADDSYMPTOM	(procedure in DEMO.PRG)
Contains: ADDSYMPTOM	(Params: MNAME, MVAL)
Called by: ADDHISTORY	(procedure in DEMO.PRG)
Called by: GETSYMPTOM()	(function in U.PRG)
Calls: DEFQ	(procedure in DEMO.PRG)
Contains: DELDIAGNOS	(Params: none)
Called by: INIDIAGNOS()	(function in DEMO.PRG)
Called by: BATCHTEST()	(function in DEMO.PRG)
Called by: M_ENCNEW	(procedure in DEMO.PRG)
Called by: LODENC	(procedure in DEMO.PRG)
Calls: RESETDIAG	(procedure in DEMO.PRG)
Contains: RESETDIAG	(Params: none)
Called by: M_ASSIST	(procedure in DEMO.PRG)
Called by: DELDIAGNOS	(procedure in DEMO.PRG)
Contains: DELSYMPTOM	(Params: none)
Called by: INISYMPTOM()	(function in DEMO.PRG)
Called by: BATCHTEST()	(function in DEMO.PRG)
Called by: M_ENCNEW	(procedure in DEMO.PRG)
Called by: LODENC	(procedure in DEMO.PRG)
Contains: INCR	(Params: MSTR, MINCR)
Called by: DECR	(procedure in DEMO.PRG)
Calls: GETFACTOR	(procedure in DEMO.PRG)
Calls: SETFACTOR	(procedure in DEMO.PRG)
Contains: DECR	(Params: MSTR, MINCR)
Calls: INCR	(procedure in DEMO.PRG)
Contains: MULFACTOR	(Params: MSTR, MVAL)
Called by: EVALCOND	(procedure in DEMO.PRG)
Calls: FACTORID	(procedure in DEMO.PRG)
Contains: SUMFACTOR	(Params: none)
Called by: NORMFACTOR	(procedure in DEMO.PRG)
Contains: NORMFACTOR	(Params: none)
Called by: MAKEDIAG	(procedure in DEMO.PRG)
Calls: SUMFACTOR	(procedure in DEMO.PRG)
Contains: GETFACTOR	(Params: MSTR)
Called by: INCR	(procedure in DEMO.PRG)
Calls: FACTORID	(procedure in DEMO.PRG)
Contains: SETFACTOR	(Params: MSTR, MVAL)
Called by: INCR	(procedure in DEMO.PRG)
Calls: DEFFACTOR	(procedure in DEMO.PRG)
Contains: DEFFACTOR	(Params: MSTR, MVAL)
Called by: ADDDIAGNOS	(procedure in DEMO.PRG)
Called by: SETFACTOR	(procedure in DEMO.PRG)
Calls: FACTORID	(procedure in DEMO.PRG)
Contains: FACTORID	(Params: MSTR)
Called by: UNDOFACTOR	(procedure in DEMO.PRG)
Called by: MULFACTOR	(procedure in DEMO.PRG)
Called by: GETFACTOR	(procedure in DEMO.PRG)
Called by: DEFFACTOR	(procedure in DEMO.PRG)
Called by: ADDFACTOR	(procedure in DEMO.PRG)
Calls: ERRMESSAGE()	(function in U.PRG)
Contains: ADDFACTOR	(Params: MSTR)
Calls: FACTORID	(procedure in DEMO.PRG)
Contains: INIDIAGNOS()	(Params: none)
Called by: DEMO.PRG	(function in U.PRG)
Calls: DISEASECNT()	(procedure in DEMO.PRG)
Calls: DELDIAGNOS	(Params: none)
Contains: INISYMPTOM()	(function in U.PRG)
Called by: DEMO.PRG	(function in U.PRG)
Calls: DICTCOUNT()	(function in U.PRG)

Calls: DELSYMPTOM
 Contains: LOOKUPQ
 Called by: CHANGEQ
 Called by: LASTQ
 Calls: DICTQ()
 Contains: LOOKUPNAM
 Called by: STRCOND
 Calls: DICTQ()
 Contains: ASK
 Called by: M_REDOQ
 Called by: EVALCOND
 Calls: EVALQ
 Contains: ISQ
 Calls: EVALQ
 Contains: VALQ
 Calls: EVALQ
 Contains: EVALQ
 Called by: ASK
 Called by: ISQ
 Called by: VALQ
 Calls: QID
 Calls: EDITSYMP
 Calls: DEFQ
 Calls: PUSHQ
 Contains: EDITSYMP
 Called by: EVALQ
 Calls: DICTQ()
 Calls: ERRMESSAGE()
 Calls: DICTQV()
 Calls: EDNUM()
 Calls: EDBOOL()
 Calls: DICTECOUNT()
 Calls: DICTQE()
 Calls: EDCHECK()
 Contains: SETQ
 Called by: EVALCOND
 Calls: DEFQ
 Contains: DEFQ
 Called by: ADDSYMPTOM
 Called by: EVALQ
 Called by: SETQ
 Calls: QID
 Contains: GETQ
 Called by: M_REDOQ
 Called by: STRNATURAL
 Called by: SFSYMP
 Calls: QID
 Contains: ERASEQ
 Called by: M_REDOQ
 Called by: ADDHISTORY
 Calls: QID
 Contains: RESTOREQ
 Called by: M_REDOQ
 Calls: QID
 Contains: PUSHQ
 Called by: EVALQ
 Contains: CHANGEQ
 Called by: M_CHANGE
 Calls: LOOKUPQ
 Calls: CHANGELIST()
 Contains: POPQ
 Called by: M_CHANGE

(procedure in DEMO.PRG)
 (Params: MID)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (function in U.PRG)
 (Params: MID)
 (procedure in DEMO.PRG)
 (function in U.PRG)
 (Params: MSTR, MQ)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: MSTR, MVAL)
 (procedure in DEMO.PRG)
 (Params: MSTR)
 (procedure in DEMO.PRG)
 (Params: MSTR)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: MSTR, MVAL)
 (procedure in DEMO.PRG)
 (function in U.PRG)
 (function in U.PRG)
 (function in U.PRG)
 (function in U.PRG)
 (function in U.PRG)
 (function in U.PRG)
 (function in U.PRG)
 (Params: MSTR, MVAL)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: MSTR, MVAL)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: MSTR)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: MSTR)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: MSTR, MVAL)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: MSTR, MVAL)
 (procedure in DEMO.PRG)
 (Params: N)
 (procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (function in U.PRG)
 (Params: N)
 (procedure in DEMO.PRG)

Called by: M UNDO	(procedure in DEMO.PRG)
Calls: UNDOQUAL	(procedure in DEMO.PRG)
Contains: LASTQ	(Params: none)
Called by: M UNDO	(procedure in DEMO.PRG)
Calls: LOOKUPQ	(procedure in DEMO.PRG)
Contains: QID	(Params: MSTR)
Called by: UNDOQUAL	(procedure in DEMO.PRG)
Called by: EVALQ	(procedure in DEMO.PRG)
Called by: DEFQ	(procedure in DEMO.PRG)
Called by: GETQ	(procedure in DEMO.PRG)
Called by: ERASEQ	(procedure in DEMO.PRG)
Called by: RESTOREQ	(procedure in DEMO.PRG)
Called by: ADDQ	(procedure in DEMO.PRG)
Calls: ERRMESSAGE()	(function in U.PRG)
Contains: ADDQ	(Params: MSTR)
Calls: QID	(procedure in DEMO.PRG)
Contains: STRCOND	(Params: MCOND, MQID, MNAME, MVAL, DELIM)
Called by: M RESUME	(procedure in DEMO.PRG)
Called by: EDITCOND	(procedure in U.PRG)
Called by: EDITHANDP	(procedure in U.PRG)
Calls: LOOKUPNAM	(procedure in DEMO.PRG)
Contains: STRDIAG	(Params: MSORTED, MDIAG, MCONF, THRESHOLD, LIMIT, SHOWCONF)
Called by: M RESUME	(procedure in DEMO.PRG)
Called by: EDITASSIS	(procedure in U.PRG)
Calls: DISEASENAM()	(function in U.PRG)
Contains: STRNATURAL	(Params: MAREA)
Calls: GETQ	(procedure in DEMO.PRG)
Calls: DICTREPORT()	(function in U.PRG)
Calls: SPLIT()	(function in U.PRG)
Contains: BATCHTEST()	(Params: TESTING, TESTFILE, LOGFILE)
Called by: DEMO.PRG	
Calls: DELSYMPTOM	(procedure in DEMO.PRG)
Calls: DELDIAGNOS	(procedure in DEMO.PRG)
Calls: M ENCRULE	(procedure in DEMO.PRG)
Calls: ENC2STR	(procedure in DEMO.PRG)
Calls: M ENCGET	(procedure in DEMO.PRG)
Calls: M ASSIST	(procedure in DEMO.PRG)
Calls: M ENCSAV	(procedure in DEMO.PRG)
Calls: ENC2BIT	(procedure in DEMO.PRG)
Calls: DISEASENAM()	(function in U.PRG)
Contains: ENC2STR	(Params: MAREA, MBIT)
Called by: BATCHTEST()	(function in DEMO.PRG)
Calls: INVERTCOND	(procedure in DEMO.PRG)
Contains: ENC2BIT	(Params: FIRE)
Called by: BATCHTEST()	(function in DEMO.PRG)
Contains: INVERTCOND	(Params: NOBJECT, MID, MOP, MVAL)
Called by: ENC2STR	(procedure in DEMO.PRG)
Calls: EVALCOND	(procedure in DEMO.PRG)
Contains: SYSINIT()	(Params: none)
Called by: DEMO.PRG	
Calls: SETPATH	(procedure in U.PRG)
Calls: GETOPTS	(procedure in U.PRG)
Contains: SYSCLOSE()	(Params: none)
Called by: DEMO.PRG	
Contains: WPAT	(Params: none)
Called by: M PAT	(procedure in DEMO.PRG)
Called by: M NEWENC	(procedure in DEMO.PRG)
Called by: M TRAINING	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: WPl.PRG	

Calls: DP.PRG	
Calls: AGEAT	(procedure in DEMO.PRG)
Calls: WPCLEAN	(procedure in DEMO.PRG)
Calls: POPHELP	(procedure in U.PRG)
Contains: WPCLEAN	(Params: none)
Called by: WPAT	(procedure in DEMO.PRG)
Contains: M SF600	(Params: none)
Calls: OPENFILES()	(function in U.PRG)
Calls: SFENCOUNT	(procedure in U.PRG)
Calls: SFREPORT	(procedure in DEMO.PRG)
Calls: FLUSHFILES()	(function in U.PRG)
Contains: SFREPORT	(Params: MAREA)
Called by: M SF600	(procedure in DEMO.PRG)
Calls: YESNO.PRG	(function in U.PRG)
Calls: POPUPSHOW()	(procedure in DEMO.PRG)
Calls: SFSYMP	(function in U.PRG)
Calls: POPUPHIDE()	(procedure in DEMO.PRG)
Calls: SFDIAG	(procedure in DEMO.PRG)
Calls: SFTREAT	(procedure in DEMO.PRG)
Calls: EDITTEXT	(procedure in DEMO.PRG)
Contains: SFSYMP	(Params: MAREA)
Called by: SFREPORT	(procedure in DEMO.PRG)
Calls: CASECONVERT	(procedure in U.PRG)
Calls: GETQ	(procedure in DEMO.PRG)
Calls: DICTREPORT()	(function in U.PRG)
Calls: DP.PRG	
Contains: EDITTEXT	(Params: MTEXT)
Called by: SFREPORT	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: WP6.SPR	
Calls: POPHELP	
Calls: WP7.SPR	(procedure in U.PRG)
Contains: SFDIAG	(Params: none)
Called by: SFREPORT	(procedure in DEMO.PRG)
Calls: DISEASENAME()	(function in U.PRG)
Contains: SFTREAT	(Params: none)
Called by: SFREPORT	(procedure in DEMO.PRG)
Contains: M NEWENC	(Params: none)
Calls: OPENFILES()	(function in U.PRG)
Calls: WPAT	(procedure in DEMO.PRG)
Calls: M ENC	(procedure in DEMO.PRG)
Calls: KEYLOG.PRG	
Calls: FLUSHFILES()	(function in U.PRG)
Contains: M TRAINING	(Params: none)
Calls: OPENFILES()	(function in U.PRG)
Calls: WPAT	(procedure in DEMO.PRG)
Calls: KEYLOG.PRG	
Calls: M ENC	(procedure in DEMO.PRG)
Calls: FLUSHFILES()	(function in U.PRG)
Contains: BACKUP	(Params: MESSAGE, WILDCARD, FILENAME)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: BACKUP.SPR	
Calls: POPHELP	
Calls: KEYLOG.PRG	(procedure in U.PRG)
Contains: RESTORE	(Params: MESSAGE, WILDCARD, FILENAME)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: RESTORE.SPR	
Calls: POPHELP	
Calls: KEYLOG.PRG	(procedure in U.PRG)
Contains: M LOOKUP	(Params: none)
Calls: DISEASECNT()	(function in U.PRG)
Calls: DISNAMES()	(function in U.PRG)

Calls: GETDISLOOKUPLIST	(procedure in U.PRG)
Calls: KEYLOG.PRG	
Contains: M DATA	(Params: none)
Calls: GETDIAGAREA	(procedure in DEMO.PRG)
Calls: QUESTION.PRG	
Calls: KEYLOG.PRG	

U.PRG -- Last updated: 10/13/92 at 10:21

Contains: WCHOICE	(Params: TOP, LEFT, BOTTOM, RIGHT, MPAD, MCOLOR, MTITLE, MHELP, MCHOICE)
Called by: CHANGELIST()	(function in U.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: W14.SPR	
Calls: POPHELP	(procedure in U.PRG)
Contains: WCONFIRM()	(Params: M.PROMPT, MHELP)
Called by: M RESUME	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: W10.SPR	
Calls: POPHELP	(procedure in U.PRG)
Contains: USERINIT()	(Params: NOTIME)
Called by: DEMO.PRG	
Calls: INITHELP	(procedure in U.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: GETCOLORS	(procedure in U.PRG)
Contains: USERCLOSE()	(Params: none)
Called by: DEMO.PRG	
Contains: CHANGELIST()	(Params: QNAMES, MCHOICE)
Called by: CHANGEQ	(procedure in DEMO.PRG)
Calls: WCHOICE	(procedure in U.PRG)
Contains: GETDISLIST()	(Params: MLIST)
Called by: ASKDIAG	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: W7.SPR	
Calls: POPHELP	(procedure in U.PRG)
Calls: DISEASEGET	(procedure in U.PRG)
Calls: W8.PRG	
Contains: GETDISLOOKUPLIST	(Params: MLIST)
Called by: M LOOKUP	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: W7A.SPR	
Calls: POPHELP	(procedure in U.PRG)
Calls: DISEASEGET	(procedure in U.PRG)
Calls: W8.PRG	
Contains: EDNUM()	(Params: MID, MNAME, MQUESTION, MEXTRA, MHI, MLO, MWIDTH, MDEC, MUNITS, MSTRVAL)
Called by: EDITSYMP	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: W6.SPR	
Calls: POPHELP	(procedure in U.PRG)
Contains: EDBOOL()	(Params: ID, MNAME, MQUESTION, MEXTRA, MSTRVAL)
Called by: EDITSYMP	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: W6C.SPR	
Calls: POPHELP	(procedure in U.PRG)
Contains: EDLIST	(Params: ID, MNAME, MQUESTION, MEXTRA, CHOICES, MSTRVAL)
Calls: PUSHHELP	(procedure in U.PRG)

Calls: W6B.SPR
 Calls: POPHELP
 Contains: EDCHECK()

Called by: EDITSYMP
 Calls: PUSHHELP
 Calls: SETCHECK
 Calls: W6A.PRG
 Calls: POPHELP
 Contains: EDCHECK
 Called By: W6A.PRG
 Calls: SETCHECK
 Contains: SETCHECK
 Called by: EDCHECK()
 Called by: EDCHECK
 Contains: GETENCLIST()
 Called by: GETENCNT
 Calls: PUSHHELP
 Calls: W11.SPR
 Calls: POPHELP
 Contains: ENCNTTITLE()

Called by: M RESUME
 Calls: PUSHHELP
 Calls: W12.SPR
 Calls: POPHELP
 Contains: EVALID
 Called By: W12.SPR
 Calls: PUSHHELP
 Calls: W2.SPR
 Calls: POPHELP
 Contains: EDITCOND
 Called By: W12.SPR
 Called by: W12A.SPR
 Calls: M REDOQ
 Calls: STRCOND
 Contains: EDITVDIAG
 Called By: W12.SPR
 Called by: W12B.SPR
 Calls: DISEASEGET
 Calls: PUSHHELP
 Calls: W8.PRG
 Calls: POPHELP
 Contains: EDITCOMPL
 Called By: W12.SPR
 Calls: M COMPL
 Calls: GETEXAREA()
 Contains: EDITDIAG
 Called By: W12.SPR
 Calls: M DIAG
 Calls: DISEASENAM()
 Contains: EDITHANDP
 Called By: W12.SPR
 Called by: W12A.SPR
 Calls: M HANDP
 Calls: STRCOND
 Contains: EDITASSIS
 Called By: W12.SPR

(procedure in U.PRG)
 (Params: ID, MNAME, MQUESTION,
 MEXTRA, CHOICES, INCLUSIVE, CHECK,
 MSTRVAL)

(procedure in DEMO.PRG)
 (procedure in U.PRG)
 (procedure in U.PRG)

(procedure in U.PRG)
 (Params: none)

(procedure in U.PRG)
 (Params: N, CHECK, INCLUSIVE)
 (function in U.PRG)
 (procedure in U.PRG)
 (Params: MLIST, MKEY)
 (procedure in DEMO.PRG)
 (procedure in U.PRG)

(procedure in U.PRG)
 (Params: ME_SSN, ME_LASTNAM,
 ME_FIRSTNA, ME_SEX, ME_AGE,
 ME_DATEENC, ME_TIMEENC, ME_AREA,
 ME_PROVIDR, ME_DIAGNOS, ME_DIAG,
 ME_COND, ME TREAT, ME STATE)
 (procedure in DEMO.PRG)
 (procedure in U.PRG)

(procedure in U.PRG)
 (Params: none)

(procedure in U.PRG)

(procedure in U.PRG)
 (Params: MITEM)

(procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: MITEM)

(procedure in U.PRG)
 (procedure in U.PRG)

(procedure in U.PRG)
 (Params: none)

(procedure in DEMO.PRG)
 (function in U.PRG)
 (Params: none)

(procedure in DEMO.PRG)
 (function in U.PRG)
 (Params: none)

(procedure in DEMO.PRG)
 (procedure in DEMO.PRG)
 (Params: none)

Called by: W12B.SPR	
Calls: M ASSIST	(procedure in DEMO.PRG)
Calls: STRDIAG	(procedure in DEMO.PRG)
Contains: EDITTREAT	(Params: none)
Called by: W12.SPR	
Calls: M TREAT	(procedure in DEMO.PRG)
Contains: RELEASE	(Params: MWINS)
Contains: PRINTENC	(Params: MPRINTFILE)
Called by: Q7IOLH8WD()	(function in W2.SPR)
Contains: STOPREAD	(Params: MWINDOW)
Called by: W12.SPR	
Called by: W4.SPR	
Contains: ERRMESSAGE()	(Params: ERRSTR)
Called by: M PATGET	(procedure in DEMO.PRG)
Called by: FACTORID	(procedure in DEMO.PRG)
Called by: EDITSYMP	(procedure in DEMO.PRG)
Called by: QID	(procedure in DEMO.PRG)
Contains: POPMESSAGE()	(Params: ERRSTR)
Called by: M ENC	(procedure in DEMO.PRG)
Called by: M COMPL	(procedure in DEMO.PRG)
Called by: M DIAG	(procedure in DEMO.PRG)
Called by: M ASSIST	(procedure in DEMO.PRG)
Called by: M HANDP	(procedure in DEMO.PRG)
Called by: M TREAT	(procedure in DEMO.PRG)
Called by: M ENCSAV	(procedure in DEMO.PRG)
Called by: M CHANGE	(procedure in DEMO.PRG)
Called by: M PATPUT	(procedure in DEMO.PRG)
Called by: M PATADD	(procedure in DEMO.PRG)
Called by: QUIT	(procedure in U.PRG)
Called by: SEARCH	(procedure in U.PRG)
Called by: MORESEARCH	(procedure in U.PRG)
Called by: SFENCOUNT	(procedure in U.PRG)
Called by: SFGETENC	(procedure in U.PRG)
Called by: Q8DOTWCZ4()	(function in WP2.PRG)
Called by: Q8DOTWDV1()	(function in WP2.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: W9.SPR	
Calls: POPHELP	(procedure in U.PRG)
Contains: POPUPSHOW()	(Params: ERRSTR)
Called by: DEMO.PRG	
Called by: OPENFILES()	(function in U.PRG)
Called by: M ASSIST	(procedure in DEMO.PRG)
Called by: M HANDP	(procedure in DEMO.PRG)
Called by: M ENCGET	(procedure in DEMO.PRG)
Called by: SFREPORT	(procedure in DEMO.PRG)
Called by: REINDEX	(procedure in U.PRG)
Contains: POPUPHIDE()	(Params: W)
Called by: DEMO.PRG	
Called by: OPENFILES()	(function in U.PRG)
Called by: M ASSIST	(procedure in DEMO.PRG)
Called by: M HANDP	(procedure in DEMO.PRG)
Called by: M ENCGET	(procedure in DEMO.PRG)
Called by: SFREPORT	(procedure in DEMO.PRG)
Called by: SEARCH	(procedure in U.PRG)
Called by: MORESEARCH	(procedure in U.PRG)
Called by: REINDEX	(procedure in U.PRG)
Contains: GETAREALIST()	(Params: MLIST)
Called by: GETDIAGAREA	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: W5.SPR	
Calls: POPHELP	(procedure in U.PRG)
Contains: GETPATLIST()	(Params: MKEY, MLIST)

Called by: GETPATIENT	(procedure in DEMO.PRG)	
Calls: PUSHHELP	(procedure in U.PRG)	
Calls: W3.SPR		
Calls: POPHELP	(procedure in U.PRG)	
Contains: MAINMENU()	(Params: none)	
Called by: DEMO.PRG		
Calls: CAMD.MPR		
Calls: MYHANDLER	(procedure in U.PRG)	
Contains: MYHANDLER	(Params: none)	
Called by: MAINMENU()	(function in U.PRG)	
Calls: M_RESUME	(procedure in DEMO.PRG)	
Calls: M_REGPAT	(procedure in DEMO.PRG)	
Contains: MHIT	(Params: PROG)	
Called by: CAMD.MPR		
Called by: Q9POPFLR8	(procedure in CAMD.MPR)	
Called by: CAMDTR.MPR		
Contains: QUIT	(Params: none)	
Called by: CAMD.MPR		
Called by: CAMDTR.MPR		
Calls: POPMESSAGE()	(function in U.PRG)	
Contains: PATID()	(Params: MPATIENTID, MSOCSEC,	
	MLASTNAME, MFIRSTNAME, MDATEBIRTH,	
	MSEX, MRATERANK, ISEDT)	
	(procedure in DEMO.PRG)	
	(procedure in U.PRG)	
Called by: M_REGPAT		
Calls: PUSHHELP		
Calls: W4.SPR		
Calls: POPHELP	(procedure in U.PRG)	
Contains: PVALID	(Params: none)	
Called by: W4.SPR		
Contains: UNDOMENU()	(Params: none)	
Called by: MYUNDO	(procedure in DEMO.PRG)	
Calls: W13.SPR		
Contains: PUSHHELP	(Params: HELPTOPIC)	
Called by: WP2.PRG		
Called by: USERINIT()	(function in U.PRG)	
Called by: WPAT	(procedure in DEMO.PRG)	
Called by: EDITTEXT	(procedure in DEMO.PRG)	
Called by: BACKUP	(procedure in DEMO.PRG)	
Called by: RESTORE	(procedure in DEMO.PRG)	
Called by: ENCNTITLE()	(function in U.PRG)	
Called by: WCONFIRM()	(function in U.PRG)	
Called by: PATID()	(function in U.PRG)	
Called by: POPMESSAGE()	(function in U.PRG)	
Called by: GETDISLIST()	(function in U.PRG)	
Called by: GETAREALIST()	(function in U.PRG)	
Called by: GETPATLIST()	(function in U.PRG)	
Called by: GETENCLIST()	(function in U.PRG)	
Called by: EDNUM()	(function in U.PRG)	
Called by: EDBOOL()	(function in U.PRG)	
Called by: EDCHECK()	(function in U.PRG)	
Called by: WCHOICE	(procedure in U.PRG)	
Called by: GETDISLOOKUPLIST	(procedure in U.PRG)	
Called by: EDLIST	(procedure in U.PRG)	
Called by: EVALID	(procedure in U.PRG)	
Called by: EDITVDIAG	(procedure in U.PRG)	
Called by: WPGETPATLIST	(procedure in U.PRG)	
Called by: SEARCH	(procedure in U.PRG)	
Called by: SFENCOUNT	(procedure in U.PRG)	
Called by: SFENCLIST	(procedure in U.PRG)	
Called by: SFGETPATLIST	(procedure in U.PRG)	
Called by: Q9POPFL0Y	(procedure in CAMD.MPR)	
Calls: SETHELP	(procedure in U.PRG)	

Contains: POPHELP	(Params: none)
Called by: WP2.PR	(procedure in DEMO.PR
Called by: WPAT	(procedure in DEMO.PR
Called by: EDITTEXT	(procedure in DEMO.PR
Called by: BACKUP	(procedure in DEMO.PR
Called by: RESTORE	(procedure in DEMO.PR
Called by: ENCNTTITLE()	(function in U.PR
Called by: WCONFIRM()	(function in U.PR
Called by: PATID()	(function in U.PR
Called by: POPMESSAGE()	(function in U.PR
Called by: GETDISLIST()	(function in U.PR
Called by: GETAREALIST()	(function in U.PR
Called by: GETPATLIST()	(function in U.PR
Called by: GETENCLIST()	(function in U.PR
Called by: EDNUM()	(function in U.PR
Called by: EDBOOL()	(function in U.PR
Called by: EDCHECK()	(function in U.PR
Called by: WCHOICE	(procedure in U.PR
Called by: GETDISLOOKUPLIST	(procedure in U.PR
Called by: EDLIST	(procedure in U.PR
Called by: EVALID	(procedure in U.PR
Called by: EDITVDIAG	(procedure in U.PR
Called by: WPGETPATLIST	(procedure in U.PR
Called by: SEARCH	(procedure in U.PR
Called by: SFENCOUNT	(procedure in U.PR
Called by: SFENCLIST	(procedure in U.PR
Called by: SFGETPATLIST	(procedure in U.PR
Called by: Q9POPFLLOY	(procedure in CAMD.MPR
Calls: SETHelp	(procedure in U.PR
Contains: SETHelp	(Params: none)
Called by: PUSHHELP	(procedure in U.PR
Called by: POPHELP	(procedure in U.PR
Contains: OLMR	(Params: none)
Called by: INITHELP	(procedure in U.PR
Contains: INITHELP	(Params: none)
Called by: USERINIT()	(function in U.PR
Calls: OLMR	(procedure in U.PR
Contains: AREACOUNT()	(Params: none)
Called by: GETDIAGAREA	(procedure in DEMO.PR
Contains: AREANAMES()	(Params: MAREA, MNAME)
Called by: GETDIAGAREA	(procedure in DEMO.PR
Contains: GETEXAREA()	(Params: MAREA)
Called by: M RESUME	(procedure in DEMO.PR
Called by: EDITCOMPL	(procedure in U.PR
Contains: GETRULEBASE()	(Params: MAREA, CTHRESHOLD, CTHRESHPRO, CTHRESHLIK, NRULES)
Called by: LODRULE	(procedure in DEMO.PR
Contains: GETCOLORS	(Params: COLORSET)
Called by: USERINIT()	(function in U.PR
Contains: PUTDIAGNOS()	(Params: MENCOUNTID, MDIAG, MCONF)
Called by: M_ENCSAV	(procedure in DEMO.PR
Calls: OPENDIAG	(procedure in U.PR
Contains: GETDIAGNOS()	(Params: MENCOUNTID)
Called by: LODENC	(procedure in DEMO.PR
Calls: OPENDIAG	(procedure in U.PR
Calls: ADDDIAGNOS	(procedure in DEMO.PR
Contains: OPENDIAG	(Params: none)
Called by: PUTDIAGNOS()	(function in U.PR
Called by: GETDIAGNOS()	(function in U.PR
Contains: OPENDICT	(Params: none)
Called by: OPENFILES()	(function in U.PR
Contains: CLOSEDICT	(Params: none)

Called by: FLUSHFILES()	(function in U.PRG)
Contains: DICTCOUNT()	(Params: none)
Called by: INISYMPOM()	(function in DEMO.PRG)
Contains: DICTQ()	(Params: MID, MASK, MQ, MNAME, MPROMPT)
Called by: LOOKUPQ	(procedure in DEMO.PRG)
Called by: LOOKUPNAM	(procedure in DEMO.PRG)
Called by: EDITSYMP	(procedure in DEMO.PRG)
Called by: PUTSYMPOM()	(function in U.PRG)
Called by: GETSYMPOM()	(function in U.PRG)
Contains: DICTQV()	(Params: MID, MHI, MLO, MWIDTH, MDEC, MUNITS)
Called by: EDITSYMP	(procedure in DEMO.PRG)
Contains: DICTECOUNT()	(Params: MID)
Called by: EDITSYMP	(procedure in DEMO.PRG)
Contains: DICTQE()	(Params: MID, NAMES, MATTRIB)
Called by: EDITSYMP	(procedure in DEMO.PRG)
Contains: DICTORD	(Params: MID, MENUM)
Called by: PUTSYMPOM()	(function in U.PRG)
Contains: DICTENUM	(Params: MID, MORD)
Called by: GETSYMPOM()	(function in U.PRG)
Contains: DICTREPORT()	(Params: MID, MENUM)
Called by: STRNATURAL	(procedure in DEMO.PRG)
Called by: SFSYMP	(procedure in DEMO.PRG)
Contains: DICTQID	(Params: MNAME)
Contains: DICTQNAME()	(Params: MNAME)
Called by: ADDHISTORY	(procedure in DEMO.PRG)
Contains: OPENENCNT	(Params: none)
Called by: OPENFILES()	(function in U.PRG)
Contains: FLUSHENCNT	(Params: none)
Called by: FLUSHFILES()	(function in U.PRG)
Contains: NEWENCNT()	(Params: MENCOUNTID, MPATIENTID, MDATEENC, MTIMEENC, MPROVIDER, MSTATION, MDEPT, MAREA, MDIAGNOSIS, MTREATMENT, MSTATE)
Called by: M_ENCSAV	(procedure in DEMO.PRG)
Contains: ENCID	(Params: MPATIENTID, MDATE, MTIME)
Contains: PUTENCNT()	(Params: MENCOUNTID, MPATIENT, MDATEENC, MTIMEENC, MPROVIDER, MSTATION, MDEPT, MAREA, MDIAGNOSIS, MTREATMENT, MSTATE)
Called by: M_ENCSAV	(procedure in DEMO.PRG)
Contains: ENCGETINFO()	(Params: MENCOUNTID, MPATIENTID, MDATEENC, MTIMEENC, MPROVIDER, MSTATION, MDEPT, MAREA, MDIAGNOSIS, MTREATMENT, MSTATE, MREPORT)
Called by: LODENC	(procedure in DEMO.PRG)
Contains: ENCCOUNT()	(Params: none)
Called by: GETENCNT	(procedure in DEMO.PRG)
Called by: SFGETENC	(procedure in U.PRG)
Contains: ENCNames()	(Params: NAMES, IDS, PATIENTS, MPATIENT, MDATE)
Called by: GETENCNT	(procedure in DEMO.PRG)
Calls: PATSSN()	(function in U.PRG)
Contains: DISEASEGET	(Params: MDISEASE)
Called by: GETDISLIST()	(function in U.PRG)
Called by: GETDISLOOKUPLIST	(procedure in U.PRG)
Called by: EDITVDIAG	(procedure in U.PRG)
Contains: DISEASECNT()	(Params: none)
Called by: INIDIAGNOS()	(function in DEMO.PRG)
Called by: ASKDIAG	(procedure in DEMO.PRG)
Called by: M_LOOKUP	(procedure in DEMO.PRG)

Contains: DISNAMES()	(Params: MDISEASES, MAREA)
Called by: ASKDIAG	(procedure in DEMO.PRG)
Called by: M_LOOKUP	(procedure in DEMO.PRG)
Contains: DISEASEPUT	(Params: MDISEASE, MHELP)
Contains: DISEASENAM()	(Params: MID, MNAME)
Called by: BATCHTEST()	(function in DEMO.PRG)
Called by: M_RESUME	(procedure in DEMO.PRG)
Called by: M_DIAG	(procedure in DEMO.PRG)
Called by: STRDIAG	(procedure in DEMO.PRG)
Called by: EDITDIAG	(procedure in U.PRG)
Contains: DISEASEID()	(Params: DNAME, DID)
Called by: M_DIAG	(procedure in DEMO.PRG)
Contains: OPENFILES()	(Params: TESTING)
Called by: DEMO.PRG	
Called by: M_PAT	(procedure in DEMO.PRG)
Called by: M_SF600	(procedure in DEMO.PRG)
Called by: M_NEWENC	(procedure in DEMO.PRG)
Called by: M_TRAINING	(procedure in DEMO.PRG)
Calls: POPUPSHOW()	(function in U.PRG)
Calls: OPENPAT	(procedure in U.PRG)
Calls: OPENENCNT	(procedure in U.PRG)
Calls: OPENDICT	(procedure in U.PRG)
Calls: OPENRULE	(procedure in U.PRG)
Calls: POPUPHIDE()	(function in U.PRG)
Contains: FLUSHFILES()	(Params: none)
Called by: M_PAT	(procedure in DEMO.PRG)
Called by: M_SF600	(procedure in DEMO.PRG)
Called by: M_NEWENC	(procedure in DEMO.PRG)
Called by: M_TRAINING	(procedure in DEMO.PRG)
Calls: FLUSHPAT	(procedure in U.PRG)
Calls: FLUSHENCNT	(procedure in U.PRG)
Calls: CLOSEDICT	(procedure in U.PRG)
Calls: CLOSERULE	(procedure in U.PRG)
Contains: CLOSEFILES	(Params: none)
Called by: BACKUP.SPR	
Called by: RESTORE.SPR	
Called by: REINDEX	(procedure in U.PRG)
Contains: OPENPAT	(Params: none)
Called by: OPENFILES()	(function in U.PRG)
Calls: FILEFIND	(procedure in U.PRG)
Calls: ERRHAND	(procedure in U.PRG)
Calls: OPENSAMS	(procedure in U.PRG)
Contains: FLUSHPAT	(Params: none)
Called by: FLUSHFILES()	(function in U.PRG)
Contains: PATSSN()	(Params: MID)
Called by: ENCOUNT	(procedure in DEMO.PRG)
Called by: GETENCNT	(procedure in DEMO.PRG)
Called by: ENCNames()	(function in U.PRG)
Called by: SFENCOUNT	(procedure in U.PRG)
Called by: SFENCLIST	(procedure in U.PRG)
Called by: SFENCNames	(procedure in U.PRG)
Contains: PATGETINFO()	(Params: MPATIENTID, MSOCSEC, MLASTNAME, MFIRSTNAME, MDATEBIRTH, MSEX, MRATERANK)
Called by: M_PATGET	(procedure in DEMO.PRG)
Contains: PATPUTINFO()	(Params: MPATIENTID, MSOCSEC, MLASTNAME, MFIRSTNAME, MDATEBIRTH, MSEX, MRATERANK)
Called by: M_PATPUT	(procedure in DEMO.PRG)
Contains: NEWPATIENT()	(Params: MPATIENTID, MSOCSEC, MLASTNAME, MFIRSTNAME, MDATEBIRTH, MSEX, MRATERANK)

Called by: M PATADD	(procedure in DEMO.PRG)
Contains: DELPATIENT	(Params: MPATIENTID)
Contains: ISVALIDPAT	(Params: MPATIENTID)
Contains: ISVALIDSSN()	(Params: MSOCSEC)
Called by: M PATPUT	(procedure in DEMO.PRG)
Called by: M PATADD	(procedure in DEMO.PRG)
Contains: PATCOUNT()	(Params: none)
Called by: GETPATIENT	(procedure in DEMO.PRG)
Contains: PATNAMES()	(Params: NAMES, DELIM, MKEY)
Called by: GETPATIENT	(procedure in DEMO.PRG)
Calls: STRNAME	(procedure in U.PRG)
Contains: STRNAME	(Params: MSOCSEC, MLASTNAM, MFIRSTNAME, DELIM)
Called by: PATNAMES()	(function in U.PRG)
Contains: OPENRULE	(Params: none)
Called by: OPENFILES()	(function in U.PRG)
Contains: CLOSERULE	(Params: none)
Called by: FLUSHFILES()	(function in U.PRG)
Contains: GETRULE	(Params: RULEID)
Contains: PUTSYMPTOM()	(Params: MENCOUNTID, MCOND, MVAL)
Called by: M ENCSAV	(procedure in DEMO.PRG)
Calls: OPENSYP	(procedure in U.PRG)
Calls: SPLIT()	(function in U.PRG)
Calls: DICTQ()	(function in U.PRG)
Calls: DICTORD	(procedure in U.PRG)
Contains: SPLIT()	(Params: MQ)
Called by: STRNATURAL	(procedure in DEMO.PRG)
Called by: PUTSYMPTOM()	(function in U.PRG)
Contains: GETSYMPTOM()	(Params: MENCOUNTID)
Called by: LODENC	(procedure in DEMO.PRG)
Calls: OPENSYP	(procedure in U.PRG)
Calls: DICTQ()	(function in U.PRG)
Calls: DICTENUM	(procedure in U.PRG)
Calls: ADDSYMPTOM	(procedure in DEMO.PRG)
Contains: OPENSYP	(Params: none)
Called by: PUTSYMPTOM()	(function in U.PRG)
Called by: GETSYMPTOM()	(function in U.PRG)
Contains: GETSTATION()	(Params: MSTATION, MDEPT)
Called by: INIENCOUNT()	(function in DEMO.PRG)
Contains: GETOPTS	(Params: IS DEBUG, IS CONF, IS RULE, IS NOVICE, IS_EDIT, MPRIVATE, MDISCLAIM)
Called by: SYSINIT()	(function in DEMO.PRG)
Contains: NEXTPAT	(Params: INC)
Called by: ADDPAT()	(function in U.PRG)
Contains: ADDPAT()	(Params: none)
Called by: M PATADD	(procedure in DEMO.PRG)
Calls: NEXTPAT	(procedure in U.PRG)
Contains: NEXTENCNT()	(Params: INC)
Called by: M ENCNEW	(procedure in DEMO.PRG)
Called by: ADDENCNT()	(function in U.PRG)
Contains: ADDENCNT()	(Params: none)
Called by: M ENCSAV	(procedure in DEMO.PRG)
Calls: NEXTENCNT()	(function in U.PRG)
Contains: NENCNT	(Params: none)
Contains: NEXTUSER	(Params: INC)
Called by: ADDUSER	(procedure in U.PRG)
Contains: ADDUSER	(Params: none)
Calls: NEXTUSER	(procedure in U.PRG)
Contains: BROWSEPAT	(Params: MSOC, MLAST, MFIRST, MMID)
Called by: Q8DOYAR6U()	(function in WPL.PRG)
Calls: ERRMSG.PRG	

Calls: WPGETPATLIST	(procedure in U.PRG)
Contains: WPGETPATLIST	(Params: MLIST)
Called by: BROWSEPAT	(procedure in U.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: WP3.SPR	
Calls: POPHELP	(procedure in U.PRG)
Calls: DP.PRG	
Calls: GETPAT	(procedure in U.PRG)
Calls: WP2.PRG	
Contains: SEARCH	(Params: MSOC, MLAST, MFIRST)
Called by: Q8DOYAR6U()	(function in WPl.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Calls: GETCAMPAT	(procedure in U.PRG)
Calls: GETSAMSPAT	(procedure in U.PRG)
Calls: DP.PRG	
Calls: GETSAMS	(procedure in U.PRG)
Calls: POPUPHIDE()	(function in U.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: WPATS.PRG	
Calls: POPHELP	(procedure in U.PRG)
Calls: WP2.PRG	
Contains: MORESEARCH	(Params: MITEN, CHOICE)
Called by: Q7IOL5TJ5()	(function in WPATS.PRG)
Calls: DP.PRG	
Calls: POPMESSAGE()	(function in U.PRG)
Calls: GETSAMSPAT	(procedure in U.PRG)
Calls: POPUPHIDE()	(function in U.PRG)
Contains: GETCAMPAT	(Params: MSSN, MLAST, MFIRST)
Called by: SEARCH	(procedure in U.PRG)
Called by: SFGETPAT	(procedure in U.PRG)
Contains: GETSAMSPAT	(Params: MSSN, MLAST, MFIRST)
Called by: SEARCH	(procedure in U.PRG)
Called by: MORESEARCH	(procedure in U.PRG)
Contains: GETPATDATA	(Params: MITEN)
Called by: RESULT	(procedure in WPATS.PRG)
Called by: PATEDIT	(procedure in WPATS.PRG)
Calls: DP.PRG	
Calls: GETPATTEMP	(procedure in U.PRG)
Calls: GETSAMS	(procedure in U.PRG)
Contains: GETSAMS	(Params: MREC)
Called by: SEARCH	(procedure in U.PRG)
Called by: GETPATDATA	(procedure in U.PRG)
Calls: SAMSDATA	(procedure in U.PRG)
Contains: SAMSDATA	(Params: none)
Called by: GETSAMS	(procedure in U.PRG)
Calls: DP.PRG	
Calls: COMBINED	(procedure in U.PRG)
Contains: GETPAT	(Params: MREC)
Called by: WPGETPATLIST	(procedure in U.PRG)
Calls: PATDATA	(procedure in U.PRG)
Contains: GETPATTEMP	(Params: MREC)
Called by: GETPATDATA	(procedure in U.PRG)
Called by: SFGETPATLIST	(procedure in U.PRG)
Calls: PATDATA	(procedure in U.PRG)
Contains: PATDATA	(Params: none)
Called by: GETPAT	(procedure in U.PRG)
Called by: GETPATTEMP	(procedure in U.PRG)
Calls: COMBINED	(procedure in U.PRG)
Contains: COMBINED	(Params: MT1, MT2, MT3, MT4, MT5, MT6, MT7, MT8, MT9)
Called by: SAMSDATA	(procedure in U.PRG)
Called by: PATDATA	(procedure in U.PRG)

Called by: SAVREC	(procedure in U.PRG)
Called by: Q8DOTWDV1()	(function in WP2.PRG)
Contains: SAVREC	(Params: MFLAG, MITEM)
Called by: Q8DOTWDV1()	(function in WP2.PRG)
Called by: RESULT	(procedure in WPATS.PRG)
Calls: DP.PRG	
Calls: COMBINED	(procedure in U.PRG)
Contains: OPENSAMS	(Params: none)
Called by: OPENPAT	(procedure in U.PRG)
Calls: SETPATH	(procedure in U.PRG)
Calls: FILEFIND	(procedure in U.PRG)
Contains: CASECONVERT	(Params: MSTR, MCASE)
Called by: SFSYMP	(procedure in DEMO.PRG)
Contains: SFENCOUNT	(Params: none)
Called by: M_SF600	(procedure in DEMO.PRG)
Calls: M_PATGET	(procedure in DEMO.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: WP4.SPR	
Calls: POPHELP	(procedure in U.PRG)
Calls: SFGETPAT	(procedure in U.PRG)
Calls: SFGETENC	(procedure in U.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Calls: PATSSN()	(function in U.PRG)
Calls: LODENC	(procedure in DEMO.PRG)
Contains: SFGETENC	(Params: PAR1, PAR2)
Called by: SFENCOUNT	(procedure in U.PRG)
Calls: ENCCOUNT()	(function in U.PRG)
Calls: SFENCNAMES	(procedure in U.PRG)
Calls: SFENCLIST	(procedure in U.PRG)
Calls: POPMESSAGE()	(function in U.PRG)
Contains: SFENCLIST	(Params: MLIST, MPAT, MDATE)
Called by: SFGETENC	(procedure in U.PRG)
Calls: PATSSN()	(function in U.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: WP5.SPR	
Calls: POPHELP	(procedure in U.PRG)
Contains: SFGETPAT	(Params: MSOC, MLASTT, MFIRSTT)
Called by: SFENCOUNT	(procedure in U.PRG)
Calls: GETCAMPAT	(procedure in U.PRG)
Calls: SFGETPATLIST	(procedure in U.PRG)
Calls: DP.PRG	
Contains: SFENCNAMES	(Params: NAMES, IDS, PATIENTS, MPATIENT, MDATE)
Called by: SFGETENC	(procedure in U.PRG)
Calls: PATSSN()	(function in U.PRG)
Contains: SFGETPATLIST	(Params: MLIST)
Called by: SFGETPAT	(procedure in U.PRG)
Calls: PUSHHELP	(procedure in U.PRG)
Calls: WP8.SPR	
Calls: POPHELP	
Calls: DP.PRG	(procedure in U.PRG)
Calls: GETPATTEMP	
Contains: SFPRINT	(procedure in U.PRG)
Called by: Q7IOLI7F7()	(Params: MPRINTFILE, MSFSTR)
Contains: SETPATH	(function in WP7.SPR)
Called by: SYSINIT()	(Params: NEWPATH)
Called by: OPENSAMS	(function in DEMO.PRG)
Called by: PKUNZIP	(procedure in U.PRG)
Contains: ERRHAND	(procedure in RESTORE.SPR)
Called by: OPENPAT	(Params: ERRMSG, QUITFLAG)
Calls: ERRMSG.PRG	(procedure in U.PRG)
Contains: FILEFIND	(Params: MFILENAME)

Called by: KEYLOG.PRG
 Called by: OPENPAT (procedure in U.PRG)
 Called by: OPENSAMS (procedure in U.PRG)
 Calls: DP.PRG
 Contains: REINDEX (Params: none)
 Called by: CAMD.MPR
 Called by: CAMDTR.MPR
 Calls: POPUPSHOW() (function in U.PRG)
 Calls: CLOSEFILES (procedure in U.PRG)
 Calls: POPUPHIDE() (function in U.PRG)

W12.SPR -- Last updated: 10/13/92 at 14:17

Contains: Q9VOUMCRG() (Params: none)
 Called by: W12.SPR
 Calls: W12C.SPR
 Contains: Q9VOUMD2M() (Params: none)
 Called by: W12.SPR
 Contains: Q9VOUMDDI() (Params: none)
 Called by: W12.SPR
 Contains: Q9VOUMDJJ() (Params: none)
 Called by: W12.SPR
 Calls: W12A.SPR
 Contains: Q9VOUMDQF() (Params: none)
 Called by: W12.SPR
 Contains: Q9VOUMDWG() (Params: none)
 Called by: W12.SPR
 Calls: W12B.SPR
 Contains: Q9VOUME80() (Params: none)
 Called by: W12.SPR

W2.SPR -- Last updated: 07/20/92 at 10:01

Contains: Q7IOLH8OT() (Params: none)
 Called by: W2.SPR
 Called by: Q2.PRG
 Contains: Q7IOLH8WD() (Params: none)
 Called by: W2.SPR
 Called by: Q2.PRG
 Calls: PRINTENC (procedure in U.PRG)
 Contains: Q7IOLH94Q() (Params: none)
 Called by: W2.SPR
 Called by: Q2.PRG

WP1.PRG -- Last updated: 08/20/92 at 16:00

Contains: Q8DOYAR6U() (Params: none)
 Called by: WP1.PRG
 Calls: SEARCH (procedure in U.PRG)
 Calls: BROWSEPAT (procedure in U.PRG)

WP2.PRG -- Last updated: 10/09/92 at 14:13

Contains: Q8DOTWCZ4() (Params: none)
 Called by: WP2.PRG
 Calls: POPMESSAGE() (function in U.PRG)
 Contains: Q8DOTWDKT() (Params: none)
 Called by: WP2.PRG

Contains: Q8DOTWDV1() (Params: none)
 Called By: WP2.PRG
 Calls: POPMESSAGE() (function in U.PRG)
 Calls: COMBINED (procedure in U.PRG)
 Calls: SAVREC (procedure in U.PR

WPATS.PRG -- Last updated: 07/31/92 at 14:41

Contains: RESULT (Params: MITEM)
 Called by: Q7IOL5TJ5() (function in WPATS.PRG)
 Calls: DP.PRG
 Calls: GETPATDATA (procedure in U.PRG)
 Calls: SAVREC (procedure in U.PRG)
 Contains: PATEDIT (Params: MITEM)
 Called by: Q7IOL5TJ5() (function in WPATS.PRG)
 Calls: DP.PRG
 Calls: GETPATDATA (procedure in U.PRG)
 Calls: WP2.PRG
 Contains: PATNEW (Params: none)
 Called by: Q7IOL5TJ5() (function in WPATS.PRG)
 Calls: WP2.PRG
 Contains: Q7IOL5TB2() (Params: none)
 Called By: WPATS.PRG
 Contains: Q7IOL5TJ5() (Params: none)
 Called By: WPATS.PRG
 Calls: RESULT (procedure in WPATS.PRG)
 Calls: PATEDIT (procedure in WPATS.PRG)
 Calls: PATNEW (procedure in WPATS.PRG)
 Calls: MORESEARCH (procedure in U.PRG)
 Contains: Q7IOL5TXD() (Params: none)
 Called By: WPATS.PRG

WP3.SPR -- Last updated: 08/03/92 at 16:06

Contains: Q7W0YICA9() (Params: none)
 Called By: WP3.SPR

WP7.SPR -- Last updated: 07/20/92 at 10:02

Contains: Q7IOLI77M() (Params: none)
 Called By: WP7.SPR
 Contains: Q7IOLI7F7() (Params: none)
 Called By: WP7.SPR
 Calls: SFPRINT (procedure in U.PRG)
 Contains: Q7IOLI7OJ() (Params: none)
 Called By: WP7.SPR

SECHECK.PRG -- Last updated: 09/03/92 at 12:01

Contains: SYSWARN() (Params: MSYSWAR)
 Called by: SECHECK.PRG
 Contains: GETSYSCODE() (Params: NWARN, NPRIVATE, MDIS, MPASS)
 Called by: SECHECK.PRG
 Calls: FILEMISSING (procedure in SECHECK.PRG)
 Contains: FILEMISSING (Params: none)
 Called by: GETSYSCODE() (function in SECHECK.PRG)
 Calls: ERRMSG.PRG
 Contains: SYSCLAIM() (Params: M.TXT)

Called by: SECHECK.PRG

W12A.SPR -- Last updated: 07/06/92 at 10:46

Contains: Q740N3U9Y() (Params: none)
Called By: W12A.SPR
Contains: Q740N3UHF() (Params: none)
Called By: W12A.SPR
Calls: W12A.SPR

W12B.SPR -- Last updated: 07/06/92 at 10:46

Contains: Q740N3XWA() (Params: none)
Called By: W12B.SPR
Contains: Q740N3Y5O() (Params: none)
Called By: W12B.SPR
Calls: W12B.SPR

W12C.SPR -- Last updated: 07/06/92 at 10:47

Contains: Q740N41W1() (Params: none)
Called By: W12C.SPR
Calls: W12C.SPR
Contains: Q740N423H() (Params: none)
Called By: W12C.SPR

W4.SPR -- Last updated: 07/28/92 at 16:30

Contains: Q7Q0ZD5UR() (Params: none)
Called By: W4.SPR
Contains: Q7Q0ZD65J() (Params: none)
Called By: W4.SPR
Calls: M_PATLIST (procedure in DEMO.PRG)

BACKUP.SPR -- Last updated: 10/09/92 at 13:00

Contains: NEWDRIVEPOP (Params: none)
Called by: BACKUP.SPR
Called by: RESTORE.SPR
Contains: NEWPATHPOP (Params: none)
Called by: BACKUP.SPR
Called by: RESTORE.SPR
Called by: _Q9R0RVE7X() (function in BACKUP.SPR)
Called by: _Q9R0RVEMQ() (function in BACKUP.SPR)
Called by: _Q9R0RVF7K() (function in BACKUP.SPR)
Called by: _Q9ROUD7U5() (function in RESTORE.SPR)
Called by: _Q9ROUD888() (function in RESTORE.SPR)
Called by: _Q9ROUD8P2() (function in RESTORE.SPR)
Contains: NEWFILEPOP (Params: none)
Called by: BACKUP.SPR
Called by: RESTORE.SPR
Called by: _Q9R0RVE7X() (function in BACKUP.SPR)
Called by: _Q9R0RVEMQ() (function in BACKUP.SPR)
Called by: _Q9R0RVF7K() (function in BACKUP.SPR)
Called by: _Q9ROUD7U5() (function in RESTORE.SPR)
Called by: _Q9ROUD888() (function in RESTORE.SPR)

Called by: Q9ROUD8P2()	(function in RESTORE.SPR)
Contains: PATHSTRING	(Params: none)
Called by: Q9RORVEMQ()	(function in BACKUP.SPR)
Called by: PKZIP	(procedure in BACKUP.SPR)
Called by: Q9ROUD888()	(function in RESTORE.SPR)
Called by: PKUNZIP	(procedure in RESTORE.SPR)
Contains: PKZIP	(Params: none)
Called by: Q9RORVEV4()	(function in BACKUP.SPR)
Calls: PATHSTRING	(procedure in BACKUP.SPR)
Contains: Q9RORVDY4()	(Params: none)
Called By: BACKJP.SPR	
Contains: Q9RORVE7X()	(Params: none)
Called By: BACKUP.SPR	
Calls: YESNO.PRG	
Calls: NEWPATHPOP	(procedure in BACKUP.SPR)
Calls: NEWFILEPOP	(procedure in BACKUP.SPR)
Contains: Q9RORVEMQ()	(Params: none)
Called By: BACKUP.SPR	
Calls: PATHSTRING	(procedure in BACKUP.SPR)
Calls: NEWPATHPOP	(procedure in BACKUP.SPR)
Calls: NEWFILEPOP	(procedure in BACKUP.SPR)
Contains: Q9RORVEV4()	(Params: none)
Called By: BACKUP.SPR	
Calls: YESNO.PRG	
Calls: PKZIP	(procedure in BACKUP.SPR)
Contains: Q9RORVF7K()	(Params: none)
Called By: BACKUP.SPR	
Calls: NEWPATHPOP	(procedure in BACKUP.SPR)
Calls: NEWFILEPOP	(procedure in BACKUP.SPR)

RESTORE.SPR -- Last updated: 10/09/92 at 14:10

Contains: NEWDRIVEPOP	(Params: none)
Called by: BACKUP.SPR	
Called by: RESTORE.SPR	
Contains: NEWPATHPOP	(Params: none)
Called by: BACKUP.SPR	
Called by: RESTORE.SPR	
Called by: Q9RORVE7X()	(function in BACKUP.SPR)
Called by: Q9RORVEMQ()	(function in BACKUP.SPR)
Called by: Q9RORVF7K()	(function in BACKUP.SPR)
Called by: Q9ROUD7U5()	(function in RESTORE.SPR)
Called by: Q9ROUD888()	(function in RESTORE.SPR)
Called by: Q9ROUD8P2()	(function in RESTORE.SPR)
Contains: NEWFILEPOP	(Params: none)
Called by: BACKUP.SPR	
Called by: RESTORE.SPR	
Called by: Q9RORVE7X()	(function in BACKUP.SPR)
Called by: Q9RORVEMQ()	(function in BACKUP.SPR)
Called by: Q9RORVF7K()	(function in BACKUP.SPR)
Called by: Q9ROUD7U5()	(function in RESTORE.SPR)
Called by: Q9ROUD888()	(function in RESTORE.SPR)
Called by: Q9ROUD8P2()	(function in RESTORE.SPR)
Contains: PATHSTRING	(Params: none)
Called by: Q9RORVEMQ()	(function in BACKUP.SPR)
Called by: PKZIP	(procedure in BACKUP.SPR)
Called by: Q9ROUD888()	(function in RESTORE.SPR)
Called by: PKUNZIP	(procedure in RESTORE.SPR)
Contains: PKUNZIP	(Params: none)
Called by: Q9ROUD8G7()	(function in RESTORE.SPR)
Calls: SETPATH	(procedure in U.PRG)

Calls: PATHSTRING (procedure in BACKUP.SPR)
 Contains: Q9ROUD7NA() (Params: none)
 Called By: RESTORE.SPR
 Contains: Q9ROUD7U5() (Params: none)
 Called By: RESTORE.SPR
 Calls: YESNO.PRG
 Calls: NEWPATHPOP (procedure in BACKUP.SPR)
 Calls: NEWFILEPOP (procedure in BACKUP.SPR)
 Contains: Q9ROUD888() (Params: none)
 Called By: RESTORE.SPR
 Calls: PATHSTRING (procedure in BACKUP.SPR)
 Calls: NEWPATHPOP (procedure in BACKUP.SPR)
 Calls: NEWFILEPOP (procedure in BACKUP.SPR)
 Contains: Q9ROUD8G7() (Params: none)
 Called By: RESTORE.SPR
 Calls: PKUNZIP (procedure in RESTORE.SPR)
 Contains: Q9ROUD8P2() (Params: none)
 Called By: RESTORE.SPR
 Calls: NEWPATHPOP (procedure in BACKUP.SPR)
 Calls: NEWFILEPOP (procedure in BACKUP.SPR)
 Contains: Q9ROUD8ZO() (Params: none)
 Called By: RESTORE.SPR

W7A.SPR -- Last updated: 08/03/92 at 16:06

Contains: Q7W0YIII2() (Params: none)
 Called By: W7A.SPR

QUESTION.PRG -- Last updated: 08/04/92 at 14:48

Contains: WRITE (Params: none)
 Called by: PRINTQUES() (function in QUESTION.PRG)
 Calls: OPENFILE (procedure in QUESTION.PRG)
 Contains: OPENFILE (Params: none)
 Called by: WRITE (procedure in QUESTION.PRG)
 Contains: PRINTQUES() (Params: MPRINTFILE)
 Called by: QUESTION.PRG
 Calls: WRITE (procedure in QUESTION.PRG)

Q2.PRG -- Last updated: 08/04/92 at 16:33

Contains: Q7I0LH8OT() (Params: none)
 Called By: W2.SPR
 Called by: Q2.PRG
 Contains: Q7I0LH8WD() (Params: none)
 Called By: W2.SPR
 Called by: Q2.PRG
 Calls: PRINTENC (procedure in U.PRG)
 Contains: Q7I0LH94Q() (Params: none)
 Called By: W2.SPR
 Called by: Q2.PRG

COLORSET.SPR -- Last updated: 10/07/92 at 11:52

Contains: Q9P0PFX2H() (Params: none)
 Called By: COLORSET.SPR

CAMD.MPR -- Last updated: 10/07/92 at 11:51

Contains: MHIT	(Params: PROG)
Contains: MEXIT	(Params: none)
Contains: Q9POPFL0Y	(Params: none)
Called By: CAMD.MPR	
Calls: PUSHHELP	(procedure in U.PRG)
Calls: PWDEDT.PRG	
Calls: POPHELP	(procedure in U.PRG)
Contains: Q9POPFLR8	(Params: none)
Called By: CAMD.MPR	
Calls: CAMDTR.MPR	
Calls: MHIT	(procedure in U.PRG)
Calls: CAMD.MPR	

CAMDTR.MPR -- Last updated: 09/28/92 at 14:22

Contains: MHIT	(Params: PROG)
Contains: MEXIT	(Params: none)

Section VIII. Program Source Code

66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

```

132 @ 5,40 GET mpath ;
133 PICTURE "g" ;
134 FROM patharray ;
135 SIZE 3,18 ;
136 DEFAULT 1 ;
137 VALID q9drvemq() ;
138 COLOR SCHEME 5,6
139
140 @ 9,45 GET maction ;
141 PICTURE "g>vT \\\BackUp;\<Cancel" ;
142 SIZE 1,8,1 ;
143 DEFAULT 1 ;
144 VALID q9drvemq4()
145
146 @ 2,1 GET mfile ;
147 FROM filearray ;
148 SIZE 11,26 ;
149 DEFAULT 1 ;
150 VALID q9drv7k() ;
151 COLOR SCHEME 6
152
153 @ 14,1 SAY "Backup file name: "
154 @ 14,19 GET mfname ;
155 SIZE 1,39 ;
156 DEFAULT " " ;
157
158 -IF NOT WVISIBLE("w_backup")
159   ACTIVATE WINDOW w_backup
160 -ENDIF
161
162 READ CYCLE MODAL
163
164 RELEASE WINDOW w_backup
165
166 #REGION 0
167 -IF m.talkstat = "ON"
168   SET TALK ON
169 -ENDIF
170 -IF m.compstat = "ON"
171   SET COMPATIBLE ON
172 -ENDIF
173
174 *
175 *
176 *
177 *
178 *
179
180 #REGION 1
181 SET DEFAULT TO (m.olddrive + m.olddpath)
182 RETURN
183
184 *****
185 *****
186 *****
187 *****
188 *****
189 *****
190 *****
191 *****
192 *****
193 *****
194 *****
195 *****
196 *****
197 *****

```

BACKUP Cleanup Code

```

198 * Create the error trap
199 m.olderror = ON( "ERROR" )
200 m.error = .F.
201 ON ERROR m.error = .T.
202
203 * Create the array of legal drive names
204 DECLARE drivearray[ 1 ]
205 drivearray[ 1 ] = "A"
206
207 * Loop from A to Z
208 m.drive = "A"
209 FOR i = 1 TO 26
210
211   * Try to switch to the drive
212   SET DEFAULT TO (m.drive)
213
214   * If it worked
215   -IF NOT m.error
216     * Add the name to the last element in the array
217     drivearray[ ALEN( DriveArray ) ] = m.drive + " "
218     * Add another element at the end of the array
219     DECLARE drivearray[ ALEN( DriveArray ) + 1 ]
220   -ENDIF
221
222   * Change to the next letter in the alphabet
223   m.drive = CHR( ASC( m.drive ) + 1 )
224
225   * Reset the error trap
226   m.error = .F.
227 -NEXT
228
229 * If the first element is empty, no drives were found
230 IF EMPTY( drivearray[ 1 ] )
231   SHOW GET mdrive disabled
232 -ELSE
233   * Drives were found so cut off the last empty element
234   * added to the array in the loop
235   DECLARE drivearray[ ALEN( DriveArray ) - 1 ]
236 -ENDIF
237
238 * Reset the error trap and return to home
239 ON ERROR m.olderror
240 SET DEFAULT TO (m.oldddefault)
241
242 * Initialize the popup variable to the element in the
243 * drive array which contains the current drive
244 mdrive = ASCAN( drivearray, m.oldddefault )
245 RETURN
246
247 *****
248 *****
249 *****
250 *****
251 *****
252 *****
253 *****
254 *****
255 *****
256 *****
257 *****
258 *****
259 *****
260 *****
261 *****
262 *****
263 *****

```

* To test for legal drives this program SET DEFAULT TO each drive
* If no error occurs the drive name will be added to an array

* Create a popup with each of the legal drive names
* The popup will be based on an array of the drive names
PRIVATE oldddefault, olderror, drivearray, ERROR

* We will change drives, so remember where we started
m.oldddefault = SET("DEFAULT")

```

264 * Get the current path string
265 m.dirstring = CURDIR()
266
267 * If we are not in the root directory
268 IF LEN( m.dirstring ) > 1
269
270 * Start parsing the path string for each directory name
271 m.dircount = 1
272
273 * Continue as long as there is a pair of slashes
274 * Surrounding the next part of the string
275 DO WHILE AT( "\", m.dirstring, m.dircount ) <> 0 AND ;
276 AT( "\", m.dirstring, m.dircount + 1 ) <> 0
277
278 * Add another element at the end of the array
279 DECLARE m.array( m.dircount + 1 )
280
281 * Cut out the next set of characters between the slashes
282 m.firstchar = AT( "\", m.dirstring, m.dircount ) + 1
283
284 m.pathlength = AT( "\", m.dirstring, m.dircount + 1 ) ;
285 - m.firstchar
286
287 * And add them to the next array element
288 patharray( m.dircount + 1 ) = SUBSTR( m.dirstring, ;
289 m.firstchar, m.pathlength )
290
291 * Look for the next directory name in the path string
292 m.dircount = m.dircount + 1
293
294
295
296
297
298 * Point the popup variable at the last element in the array
299 mpath = ALEN( patharray )
300 RETURN
301
302 *****
303 PROCEDURE newfilepop
304 *****
305
306 * Create a popup with all of the file names and its subdirection in t
307 he current directory
308 * This will be based on an array as well
309
310 * Create an array with all the files matching the current wild card
311 * ADIR() creates an array with 5 columns.
312
313 PRIVATE startsort
314 * Fill an array with directory names only
315 =ADIR( dirarray, "\", "d")
316 SIZE = ALEN( dirarray, 1 )
317 IF dirarray( 1, 1 ) = "."
318 =ADEL( dirarray, 1 )
319 SIZE = SIZE - 1
320 DECLARE dirarray( size, 5 )
321
322 * We must be in the root directory "\"
323 * Add one more row to the directory array
324 SIZE = SIZE + 1
325 DECLARE dirarray( size, 5 )
326
327 * Push all the rows down by one
328 =AINS( dirarray, 1 )

```

```

329
330 * Fill in the first directory name in the array
331 * a bug in the popups makes it refuse to display a
332 * prompt of "\", use "\\" to make one appear
333 * even then the bar will automatically be non-selectable
334 dirarray( 1, 1 ) = "\\"
335
336 * Start the sort after the root name
337
338
339
340 IF ALEN( dirarray, 1 ) > 2
341
342 * Sort the array starting at the starting row
343 =ASORT( dirarray, ALEN( dirarray, 2, 1 ) )
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394

```

```

395 PROCEDURE pkzip
396 *****
397
398 PRIVATE CURDIR, datedir, msel
399 CURDIR = FULLPATH(CURDIR())
400 IF EMPTY(GETENV("CAMD"))
401   datedir = FULLPATH(GETENV("CAMD"))
402 ELSE
403   datedir = CURDIR()
404 ENDIF
405 desdir = pathstring() + mfname
406 SET DEFA TO (datedir)
407 msel = SELECT()
408 SELECT 0
409 USE backup AGAIN
410 COPY TO t0000000.txt TYPE SDF
411 IF EMPTY(GETENV("CAMDUTIL"))
412   pkzipcom = "1" + GETENV("CAMDUTIL") + "\PKZIP &desdir @t00000000.txt"
413 ELSE
414   pkzipcom = "1PKZIP &desdir @t00000000.txt"
415 ENDIF
416 @pkzipcom
417 USE
418 DELETE FILE t0000000.txt
419 SET DEFA TO (CURDIR)
420 RETURN
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459

```

Function Origin: mdrive WHEN

From Screen: BACKUP, mdrive WHEN Clause

Variable: mdrive

Called By: WHEN Clause

Object Type: Popup

Snippet Number: 1

Function Origin: mdrive VALID

From Screen: BACKUP, mdrive VALID Clause

Variable: mdrive

Called By: VALID Clause

Object Type: Popup

Snippet Number: 2

```

460 m.newdrive = drivearray(mdrive)
461
462 IF UPPER(m.newdrive) $ "A:B:"
463   mready = yesno("Please insert disk into drive " + m.newdrive, "rea
464   => dy", "Cancel")
465 ELSE
466   mready = .T.
467 ENDIF
468 IF mready
469   "Go there and reset all the other popups to match
470   SET DEFAULT TO (m.newdrive)
471   DO newpathpop
472   DO newfilepop
473 ELSE
474   mdrive = m.prevedrive
475   m.newdrive = drivearray(mdrive)
476   SHOW GETS
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523

```

Function Origin: mpath VALID

From Screen: BACKUP, mpath VALID Clause

Variable: mpath

Called By: VALID Clause

Object Type: Popup

Snippet Number: 3

Function Origin: maction VALID

From Screen: BACKUP, maction VALID Clause

Variable: maction

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 4

Function Origin: maction VALID

From Screen: BACKUP, maction VALID Clause

Variable: maction

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 7

```

_G9RORVF7K      mfile VALID      8
Function Origin:

From Screen:
Variable:      mfile
Called By:     VALID Clause
Object Type:   List
Snippet Number: 5

```


265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330

```

@ 3,0 SAY " " = COLOR W+/N
@ 4,0 SAY " " = COLOR (colorvar)
--CASE SUBSTR(instring,curlet,1) = "k"
  DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
  FROM startx,curlet TO startx + 5,curlet + 1 ;
  NONE MOSHADOW
  ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
  curlet = curlet + 8
  @ 0,0 SAY " " = COLOR (colorvar)
  @ 0,2 SAY " " = COLOR (colorvar)
  @ 1,0 SAY " " = COLOR W+/N
  @ 1,5 SAY " " = COLOR (colorvar)
  @ 2,0 SAY " " = COLOR W+/N
  @ 2,4 SAY " " = COLOR (colorvar)
  @ 3,0 SAY " " = COLOR W+/N
  @ 3,3 SAY " "
  @ 4,0 SAY " " = COLOR (colorvar)
--CASE SUBSTR(instring,curlet,1) = "u"
  DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
  FROM startx,curlet TO startx + 5,curlet + 6 ;
  NONE MOSHADOW
  ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
  curlet = curlet + 7
  @ 1,0 SAY " " = COLOR (colorvar)
  @ 1,2 SAY " " = COLOR (colorvar)
  @ 1,3 SAY " " = COLOR (colorvar)
  @ 1,5 SAY " " = COLOR (colorvar)
  @ 2,0 SAY " " = COLOR W+/N
  @ 3,0 SAY " " = COLOR W+/N
  @ 3,2 SAY " " = COLOR W+/N
  @ 3,3 SAY " " = COLOR W+/N
  @ 3,4 SAY " " = COLOR W+/N
  @ 4,0 SAY " " = COLOR (colorvar)
--CASE SUBSTR(instring,curlet,1) = "b"
  DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
  FROM startx,curlet TO startx + 5,curlet + 6 ;
  NONE MOSHADOW
  ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
  curlet = curlet + 7
  @ 0,0 SAY " " = COLOR (colorvar)
  @ 0,2 SAY " " = COLOR (colorvar)
  @ 1,0 SAY " " = COLOR W+/N
  @ 1,4 SAY " " = COLOR W+/N
  @ 2,0 SAY " " = COLOR W+/N
  @ 2,3 SAY " " = COLOR (colorvar)
  @ 2,5 SAY " " = COLOR (colorvar)
  @ 3,0 SAY " " = COLOR W+/N
  @ 4,0 SAY " " = COLOR (colorvar)
--CASE SUBSTR(instring,curlet,1) = "c"
  DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
  FROM startx,curlet TO startx + 5,curlet + 5 ;
  NONE MOSHADOW
  ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))

```

331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396

```

curlet = curlet + 6
@ 1,0 SAY " " = COLOR W+/N
@ 1,2 SAY " " = COLOR (colorvar)
@ 1,4 SAY " " = COLOR W+/N
@ 2,0 SAY " " = COLOR W+/N
@ 2,3 SAY " " = COLOR (colorvar)
@ 3,0 SAY " " = COLOR W+/N
@ 3,2 SAY " " = COLOR W+/N
@ 3,4 SAY " " = COLOR (colorvar)
@ 4,0 SAY " " = COLOR (colorvar)
--CASE SUBSTR(instring,curlet,1) = "d"
  DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
  FROM startx,curlet TO startx + 5,curlet + 5 ;
  NONE MOSHADOW
  ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
  curlet = curlet + 7
  @ 0,0 SAY " " = COLOR (colorvar)
  @ 0,5 SAY " " = COLOR (colorvar)
  @ 1,0 SAY " " = COLOR W+/N
  @ 1,2 SAY " " = COLOR W+/N
  @ 2,0 SAY " " = COLOR W+/N
  @ 2,2 SAY " " = COLOR (colorvar)
  @ 2,3 SAY " " = COLOR W+/N
  @ 3,0 SAY " " = COLOR (colorvar)
  @ 3,5 SAY " " = COLOR (colorvar)
  @ 4,0 SAY " " = COLOR (colorvar)
--CASE SUBSTR(instring,curlet,1) = "q"
  DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
  FROM startx,curlet TO startx + 5,curlet + 5 ;
  NONE MOSHADOW
  ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
  curlet = curlet + 7
  @ 1,0 SAY " " = COLOR W+/N
  @ 1,2 SAY " " = COLOR (colorvar)
  @ 1,3 SAY " " = COLOR (colorvar)
  @ 1,5 SAY " " = COLOR (colorvar)
  @ 2,0 SAY " " = COLOR W+/N
  @ 2,2 SAY " " = COLOR (colorvar)
  @ 2,3 SAY " " = COLOR W+/N
  @ 3,0 SAY " " = COLOR (colorvar)
  @ 3,5 SAY " " = COLOR (colorvar)
  @ 4,0 SAY " " = COLOR (colorvar)
  @ 4,3 SAY " " = COLOR W+/N
  @ 5,0 SAY " " = COLOR (colorvar)
--CASE SUBSTR(instring,curlet,1) = "g"
  DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
  FROM startx,curlet TO startx + 5,curlet + 6 ;
  NONE MOSHADOW
  ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
  curlet = curlet + 7
  @ 1,0 SAY " " = COLOR W+/N
  @ 1,2 SAY " " = COLOR W+/N
  @ 1,3 SAY " " = COLOR (colorvar)
  @ 1,5 SAY " " = COLOR (colorvar)

```



```

529 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
530 current = current + 7
531 @ 1,0 SAY " " COLOR (colorvar)
532 @ 1,2 SAY " " COLOR (colorvar)
533 @ 1,3 SAY " " COLOR (colorvar)
534 @ 1,5 SAY " " COLOR (colorvar)
535
536 @ 2,0 SAY " " COLOR (colorvar)
537 @ 2,1 SAY " " COLOR W/N
538
539 @ 3,0 SAY " " COLOR W/N
540 @ 3,3 SAY " " COLOR W/N
541
542 @ 4,0 SAY " " COLOR (colorvar)
543
544 CASE SUBSTR(instring,curlet,1) = "M"
545 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
546 FROM startx,current TO startx + 5,current + 4 ;
547 NONE NOSHADOW
548 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
549 current = current + 6
550 @ 0,0 SAY " " COLOR (colorvar)
551 @ 0,4 SAY " " COLOR (colorvar)
552
553 @ 1,0 SAY " " COLOR (colorvar)
554 @ 1,3 SAY " " COLOR (colorvar)
555
556 @ 2,0 SAY " " COLOR (colorvar)
557 @ 2,2 SAY " " COLOR (colorvar)
558
559 @ 3,0 SAY " " COLOR (colorvar)
560 @ 3,1 SAY " " COLOR (colorvar)
561
562 @ 4,0 SAY " " COLOR (colorvar)
563
564 CASE SUBSTR(instring,curlet,1) = "M"
565 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
566 FROM startx,current TO startx + 5,current + 6 ;
567 NONE NOSHADOW
568 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
569 current = current + 7
570 @ 1,0 SAY " " COLOR W/N
571 @ 1,1 SAY " " COLOR (colorvar)
572 @ 1,5 SAY " " COLOR (colorvar)
573
574 @ 2,0 SAY " " COLOR W/N
575 @ 2,3 SAY " " COLOR (colorvar)
576 @ 2,5 SAY " " COLOR (colorvar)
577
578 @ 3,0 SAY " " COLOR W/N
579 @ 3,4 SAY " " COLOR W/N
580
581 @ 4,0 SAY " " COLOR (colorvar)
582 startx = startx + 6
583 current = startx
584 CASE SUBSTR(instring,curlet,1) = "M"
585 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
586 FROM startx,current TO startx + 5,current + 7 ;
587 NONE NOSHADOW
588 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
589 current = current + 9
590 @ 0,2 SAY " " COLOR W/N
591 @ 0,5 SAY " " COLOR (colorvar)
592
593 @ 1,1 SAY " " COLOR W/N
594 @ 1,6 SAY " " COLOR (colorvar)
595

```

```

596
597 @ 2,0 SAY " " COLOR W/N
598 @ 2,1 SAY " " COLOR W/N
599
600 @ 3,0 SAY " " COLOR W/N
601 @ 3,3 SAY " " COLOR (colorvar)
602 @ 3,5 SAY " " COLOR W/N
603
604 @ 4,0 SAY " " COLOR (colorvar)
605
606 CASE SUBSTR(instring,curlet,1) = "M"
607 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
608 FROM startx,current TO startx + 5,current + 6 ;
609 NONE NOSHADOW
610 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
611 current = current + 8
612 @ 0,0 SAY " " COLOR W/N
613 @ 0,4 SAY " " COLOR W/N
614
615 @ 1,0 SAY " " COLOR W/N
616 @ 1,3 SAY " " COLOR W/N
617 @ 1,5 SAY " " COLOR W/N
618
619 @ 2,0 SAY " " COLOR W/N
620 @ 2,3 SAY " " COLOR (colorvar)
621 @ 2,4 SAY " " COLOR W/N
622 @ 2,6 SAY " " COLOR (colorvar)
623
624 @ 3,0 SAY " " COLOR W/N
625 @ 3,3 SAY " " COLOR W/N
626 @ 3,5 SAY " " COLOR W/N
627
628 @ 4,0 SAY " " COLOR (colorvar)
629
630 CASE SUBSTR(instring,curlet,1) = "M"
631 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
632 FROM startx,current TO startx + 5,current + 8 ;
633 NONE NOSHADOW
634 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
635 current = current + 9
636 @ 0,0 SAY " " COLOR W/N
637 @ 0,3 SAY " " COLOR W/N
638 @ 0,6 SAY " "
639
640 @ 1,0 SAY " " COLOR W/N
641 @ 1,3 SAY " " COLOR (colorvar)
642 @ 1,4 SAY " "
643 @ 1,7 SAY " " COLOR (colorvar)
644
645 @ 2,0 SAY " " COLOR W/N
646 @ 2,3 SAY " "
647 @ 3,0 SAY " " COLOR W/N
648 @ 3,6 SAY " " COLOR W/N
649
650 @ 4,0 SAY " " COLOR (colorvar)
651
652 CASE SUBSTR(instring,curlet,1) = "M"
653 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
654 FROM startx,current TO startx + 5,current + 8 ;
655 NONE NOSHADOW
656 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
657 current = current + 8
658 @ 0,0 SAY " " COLOR W/N
659 @ 0,4 SAY " "
660
661 @ 1,0 SAY " " COLOR W/N
662 @ 1,3 SAY " " COLOR W/N
663 @ 1,6 SAY " " COLOR (colorvar)
664

```


793	0 0,2 SAY " " COLOR (colorvar)	859	0 2,0 SAY " " COLOR W+/N
794	0 0,3 SAY " " COLOR W+/N	860	0 2,3 SAY " " COLOR W+/N
795	0 0,5 SAY " " COLOR (colorvar)	861	0 2,4 SAY " " COLOR W+/N
796	0 1,0 SAY " " COLOR W+/N	862	0 3,0 SAY " " COLOR W+/N
797	0 1,5 SAY " " COLOR (colorvar)	863	0 3,3 SAY " " COLOR W+/N
798	0 2,0 SAY " " COLOR W+/N	864	0 3,7 SAY " " COLOR W+/N
799	0 2,4 SAY " " COLOR W+/N	865	0 4,0 SAY " " COLOR (colorvar)
800	0 3,0 SAY " " COLOR W+/N	866	-CASE SUBSTR(INSTRING,CURLET,1) = "Q"
801	0 3,3 SAY " " COLOR (colorvar)	867	DEFINE WINDOW ("letter" + ALLTRIM(STR(CURLET))) ;
802	0 4,0 SAY " " COLOR (colorvar)	868	FROM STARTX,CURRENTY TO STARTX + 7,CURRENTY + 7 ;
803	0 0,0 SAY " " COLOR (colorvar)	869	NONE NOSHADON
804	0 0,2 SAY " " COLOR (colorvar)	870	ACTIVATE WINDOW ("letter" + ALLTRIM(STR(CURLET)))
805	0 0,5 SAY " " COLOR (colorvar)	871	CURRENTY = CURRENTY + 9
806	0 1,0 SAY " " COLOR W+/N	872	0 0,0 SAY " " COLOR W+/N
807	0 1,5 SAY " " COLOR (colorvar)	873	0 0,3 SAY " " COLOR W+/N
808	0 2,0 SAY " " COLOR W+/N	874	0 0,4 SAY " " COLOR W+/N
809	0 2,4 SAY " " COLOR W+/N	875	0 1,0 SAY " " COLOR W+/N
810	0 3,0 SAY " " COLOR W+/N	876	0 1,3 SAY " " COLOR (colorvar)
811	0 3,3 SAY " " COLOR (colorvar)	877	0 1,5 SAY " " COLOR (colorvar)
812	0 4,0 SAY " " COLOR (colorvar)	878	0 1,7 SAY " " COLOR (colorvar)
813	0 0,0 SAY " " COLOR (colorvar)	879	0 2,0 SAY " " COLOR W+/N
814	0 0,2 SAY " " COLOR (colorvar)	880	0 2,3 SAY " " COLOR (colorvar)
815	0 0,5 SAY " " COLOR (colorvar)	881	0 2,7 SAY " " COLOR (colorvar)
816	0 1,0 SAY " " COLOR W+/N	882	0 3,0 SAY " " COLOR W+/N
817	0 1,5 SAY " " COLOR (colorvar)	883	0 3,5 SAY " " COLOR W+/N
818	0 2,0 SAY " " COLOR W+/N	884	0 3,7 SAY " " COLOR (colorvar)
819	0 2,4 SAY " " COLOR W+/N	885	0 4,0 SAY " " COLOR (colorvar)
820	0 3,0 SAY " " COLOR W+/N	886	-CASE SUBSTR(INSTRING,CURLET,1) = "P"
821	0 3,3 SAY " " COLOR (colorvar)	887	DEFINE WINDOW ("letter" + ALLTRIM(STR(CURLET))) ;
822	0 4,0 SAY " " COLOR (colorvar)	888	FROM STARTX,CURRENTY TO STARTX + 5,CURRENTY + 5 ;
823	0 0,0 SAY " " COLOR (colorvar)	889	NONE NOSHADON
824	0 0,2 SAY " " COLOR (colorvar)	890	ACTIVATE WINDOW ("letter" + ALLTRIM(STR(CURLET)))
825	0 0,5 SAY " " COLOR (colorvar)	891	CURRENTY = CURRENTY + 8
826	0 1,0 SAY " " COLOR W+/N	892	0 0,0 SAY " " COLOR W+/N
827	0 1,5 SAY " " COLOR (colorvar)	893	0 0,3 SAY " " COLOR W+/N
828	0 2,0 SAY " " COLOR W+/N	894	0 0,4 SAY " " COLOR W+/N
829	0 2,4 SAY " " COLOR W+/N	895	0 1,0 SAY " " COLOR W+/N
830	0 3,0 SAY " " COLOR W+/N	896	0 1,3 SAY " " COLOR (colorvar)
831	0 3,3 SAY " " COLOR (colorvar)	897	0 1,5 SAY " " COLOR (colorvar)
832	0 4,0 SAY " " COLOR (colorvar)	898	0 1,7 SAY " " COLOR (colorvar)
833	0 0,0 SAY " " COLOR (colorvar)	899	0 2,0 SAY " " COLOR W+/N
834	0 0,2 SAY " " COLOR (colorvar)	900	0 2,3 SAY " " COLOR (colorvar)
835	0 0,5 SAY " " COLOR (colorvar)	901	0 2,7 SAY " " COLOR (colorvar)
836	0 1,0 SAY " " COLOR W+/N	902	0 3,0 SAY " " COLOR W+/N
837	0 1,5 SAY " " COLOR (colorvar)	903	0 3,3 SAY " " COLOR (colorvar)
838	0 2,0 SAY " " COLOR W+/N	904	0 3,7 SAY " " COLOR W+/N
839	0 2,4 SAY " " COLOR W+/N	905	0 4,0 SAY " " COLOR (colorvar)
840	0 3,0 SAY " " COLOR W+/N	906	-CASE SUBSTR(INSTRING,CURLET,1) = "Q"
841	0 3,3 SAY " " COLOR (colorvar)	907	DEFINE WINDOW ("letter" + ALLTRIM(STR(CURLET))) ;
842	0 4,0 SAY " " COLOR (colorvar)	908	FROM STARTX,CURRENTY TO STARTX + 5,CURRENTY + 5 ;
843	0 0,0 SAY " " COLOR (colorvar)	909	NONE NOSHADON
844	0 0,2 SAY " " COLOR (colorvar)	910	ACTIVATE WINDOW ("letter" + ALLTRIM(STR(CURLET)))
845	0 0,5 SAY " " COLOR (colorvar)	911	CURRENTY = CURRENTY + 9
846	0 1,0 SAY " " COLOR W+/N	912	0 0,0 SAY " " COLOR W+/N
847	0 1,5 SAY " " COLOR (colorvar)	913	0 0,3 SAY " " COLOR W+/N
848	0 2,0 SAY " " COLOR W+/N	914	0 0,4 SAY " " COLOR W+/N
849	0 2,4 SAY " " COLOR W+/N	915	0 1,0 SAY " " COLOR W+/N
850	0 3,0 SAY " " COLOR W+/N	916	0 1,3 SAY " " COLOR (colorvar)
851	0 3,3 SAY " " COLOR (colorvar)	917	0 1,5 SAY " " COLOR (colorvar)
852	0 4,0 SAY " " COLOR (colorvar)	918	0 1,7 SAY " " COLOR (colorvar)
853	0 0,0 SAY " " COLOR (colorvar)	919	0 2,0 SAY " " COLOR W+/N
854	0 0,2 SAY " " COLOR (colorvar)	920	0 2,3 SAY " " COLOR (colorvar)
855	0 0,5 SAY " " COLOR (colorvar)	921	0 2,7 SAY " " COLOR (colorvar)
856	0 1,0 SAY " " COLOR W+/N	922	0 3,0 SAY " " COLOR W+/N
857	0 1,5 SAY " " COLOR (colorvar)	923	0 3,3 SAY " " COLOR (colorvar)
858	0 1,7 SAY " " COLOR (colorvar)	924	0 3,7 SAY " " COLOR W+/N

```

925 @ 1,5 SAY " " COLOR (colorvar)
926 @ 1,7 SAY " " COLOR (colorvar)
927
928 @ 2,0 SAY " " COLOR W+/N
929 @ 2,3 SAY " " COLOR (colorvar)
930 @ 2,7 SAY " " COLOR (colorvar)
931
932 @ 3,0 SAY " " COLOR (colorvar)
933 @ 3,6 SAY " " COLOR (colorvar)
934
935 @ 4,0 SAY " " COLOR (colorvar)
936 @ 4,6 SAY " " COLOR (colorvar)
937 @ 4,7 SAY " " COLOR (colorvar)
938
939 -CASE SUBSTR(instring,curlet,1) = "q"
940 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
941 FROM startx,curynto TO startx + 5,curynto + 7 ;
942 NONE NOSHADOW
943 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
944 curynty = curynty + 8
945 @ 0,0 SAY " " COLOR W+/N
946 @ 0,3 SAY " " COLOR W+/N
947
948 @ 1,0 SAY " " COLOR W+/N
949 @ 1,3 SAY " " COLOR (colorvar)
950 @ 1,6 SAY " " COLOR (colorvar)
951
952 @ 2,0 SAY " " COLOR W+/N
953 @ 2,6 SAY " " COLOR (colorvar)
954
955 @ 3,0 SAY " " COLOR W+/N
956 @ 3,3 SAY " " COLOR (colorvar)
957 @ 3,6 SAY " " COLOR (colorvar)
958
959 @ 4,0 SAY " " COLOR (colorvar)
960 -CASE SUBSTR(instring,curlet,1) = "s"
961 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
962 FROM startx,curynto TO startx + 5,curynto + 7 ;
963 NONE NOSHADOW
964 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
965 curynty = curynty + 8
966 @ 0,0 SAY " " COLOR W+/N
967 @ 0,5 SAY " " COLOR W+/N
968
969 @ 1,0 SAY " " COLOR (colorvar)
970 @ 1,5 SAY " " COLOR (colorvar)
971
972 @ 2,0 SAY " " COLOR (colorvar)
973 @ 2,6 SAY " " COLOR (colorvar)
974
975 @ 3,0 SAY " " COLOR W+/N
976 @ 3,5 SAY " " COLOR W+/N
977
978 @ 4,0 SAY " " COLOR (colorvar)
979 -CASE SUBSTR(instring,curlet,1) = "t"
980 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
981 FROM startx,curynto TO startx + 5,curynto + 7 ;
982 NONE NOSHADOW
983 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
984 curynty = curynty + 8
985 @ 0,0 SAY " " COLOR W+/N
986 @ 0,6 SAY " " COLOR (colorvar)
987
988 @ 1,2 SAY " " COLOR W+/N
989
990
991
992
993
994
995
996
997
998
999
1000
1001 @ 2,2 SAY " " COLOR W+/N
1002 @ 3,2 SAY " " COLOR W+/N
1003
1004 @ 4,0 SAY " " COLOR (colorvar)
1005 -CASE SUBSTR(instring,curlet,1) = "u"
1006 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1007 FROM startx,curynto TO startx + 5,curynto + 6 ;
1008 NONE NOSHADOW
1009 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1010 curynty = curynty + 8
1011 @ 0,0 SAY " " COLOR (colorvar)
1012 @ 0,2 SAY " " COLOR (colorvar)
1013 @ 0,4 SAY " " COLOR (colorvar)
1014 @ 0,6 SAY " " COLOR (colorvar)
1015
1016 @ 1,0 SAY " " COLOR W+/N
1017 @ 1,3 SAY " " COLOR (colorvar)
1018 @ 1,6 SAY " " COLOR (colorvar)
1019
1020 @ 2,0 SAY " " COLOR W+/N
1021 @ 2,3 SAY " " COLOR W+/N
1022 @ 2,4 SAY " " COLOR W+/N
1023
1024 @ 3,0 SAY " " COLOR W+/N
1025 @ 3,2 SAY " " COLOR W+/N
1026 @ 3,3 SAY " " COLOR W+/N
1027 @ 3,4 SAY " " COLOR W+/N
1028
1029 @ 4,0 SAY " " COLOR (colorvar)
1030 -CASE SUBSTR(instring,curlet,1) = "v"
1031 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1032 FROM startx,curynto TO startx + 5,curynto + 7 ;
1033 NONE NOSHADOW
1034 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1035 curynty = curynty + 9
1036 @ 0,0 SAY " " COLOR (colorvar)
1037 @ 0,2 SAY " " COLOR (colorvar)
1038 @ 0,3 SAY " " COLOR (colorvar)
1039 @ 0,7 SAY " " COLOR (colorvar)
1040
1041 @ 1,0 SAY " " COLOR (colorvar)
1042 @ 1,3 SAY " " COLOR W+/N
1043 @ 1,4 SAY " " COLOR W+/N
1044 @ 1,7 SAY " " COLOR (colorvar)
1045
1046 @ 2,0 SAY " " COLOR W+/N
1047 @ 2,3 SAY " " COLOR W+/N
1048 @ 3,0 SAY " " COLOR W+/N
1049 @ 3,4 SAY " " COLOR W+/N
1050
1051 @ 4,0 SAY " " COLOR (colorvar)
1052 -CASE SUBSTR(instring,curlet,1) = "w"
1053 DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
1054 FROM startx,curynto TO startx + 5,curynto + 10 ;
1055 NONE NOSHADOW
1056 ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
1057 curynty = curynty + 12
1058 @ 0,0 SAY " " COLOR (colorvar)
1059 @ 0,2 SAY " " COLOR (colorvar)
1060 @ 0,4 SAY " " COLOR (colorvar)
1061 @ 0,6 SAY " " COLOR (colorvar)
1062 @ 0,8 SAY " " COLOR (colorvar)
1063 @ 0,10 SAY " " COLOR (colorvar)
1064

```



```

1123 cumenty = cumenty + 8
1124 @ 0,0 SAY " " COLOR W+/-N
1125 @ 0,1 SAY " " COLOR (colorvar)
1126 @ 0,6 SAY " " COLOR (colorvar)
1127
1128 @ 1,0 SAY " "
1129 @ 1,4 SAY " " COLOR W+/-N
1130 @ 1,6 SAY " " COLOR (colorvar)
1131
1132 @ 2,0 SAY " "
1133 @ 2,3 SAY " " COLOR W+/-N
1134
1135 @ 3,0 SAY " " COLOR W+/-N
1136 @ 3,4 SAY " "
1137
1138 @ 4,0 SAY " " COLOR (colorvar)
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1170

```

cumenty = cumenty + 8
 @ 0,0 SAY " " COLOR W+/-N
 @ 0,1 SAY " " COLOR (colorvar)
 @ 0,6 SAY " " COLOR (colorvar)
 @ 1,0 SAY " "
 @ 1,4 SAY " " COLOR W+/-N
 @ 1,6 SAY " " COLOR (colorvar)
 @ 2,0 SAY " "
 @ 2,3 SAY " " COLOR W+/-N
 @ 3,0 SAY " " COLOR W+/-N
 @ 3,4 SAY " "
 @ 4,0 SAY " " COLOR (colorvar)
 END CASE
 curlet = curlet + 1
 ENDDO

DEFINE WINDOW ("WRN") ;
 FROM x_coor*startx,y_coor-12 TO x_coor*startx + 11,y_coor + 60 ;
 NONE NOSHADOW
 ACTIVATE WINDOW ("WRN")
 @ 3,12 SAY "(c) Naval Health Research Center, Pat.Pend."
 @ 5,0 SAY "*****"
 @ 6,0 SAY "If any problem please contact: D.M. Pearsall (Des)
 Naval Health Research Ce
 P.O.Box 85122
 San Diego, CA 92186-5122
 (619) 553 - 8401
 *****"
 @ 11,0 SAY "*****"
 *****"
 WAIT "TIMEOUT of t
 IF NOT EMPTY(instring)
 curlet = 1
 FOR i = 1 TO len str
 DEACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
 RELEASE WINDOW ("letter" + ALLTRIM(STR(curlet)))
 curlet = curlet + 1
 NEXT
 ENDDIF
 DEACTIVATE WINDOW "wrn"
 RELEASE WINDOW "wrn"
 CLEAR
 *: EOF: BIGCHARS.ACT

```

@ 1,0 SAY " " COLOR W/H
@ 1,2 SAY " " COLOR W/H
@ 1,3 SAY " " COLOR W/H
@ 1,6 SAY " " COLOR (colorvar)
@ 1,7 SAY " " COLOR W/H
@ 1,9 SAY " " COLOR W/H

@ 2,1 SAY " " COLOR W/H

@ 3,2 SAY " " COLOR W/H
@ 3,6 SAY " " COLOR W/H
@ 3,9 SAY " " COLOR (colorvar)

@ 4,0 SAY " " COLOR (colorvar)
--CASE SUBSTR(instrng,curlet,1) = "w"
DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
FROM startx,curynto startx + 5,curynto + 8 ;
NONE NOSHOW
ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
curynto = curynto + 9
@ 0,0 SAY " "
@ 0,2 SAY " " COLOR (colorvar)
@ 0,3 SAY " "
@ 0,7 SAY " " COLOR (colorvar)

@ 1,0 SAY " "
@ 1,5 SAY " " COLOR W/H
@ 1,6 SAY " " COLOR (colorvar)

@ 2,0 SAY " "
@ 2,3 SAY " " COLOR W/H

@ 3,0 SAY " " COLOR W/H
@ 3,3 SAY " " COLOR (colorvar)
@ 3,4 SAY " "
@ 3,7 SAY " " COLOR (colorvar)

@ 4,0 SAY " " COLOR (colorvar)
--CASE SUBSTR(instrng,curlet,1) = "w"
DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
FROM startx,curynto startx + 5,curynto + 9 ;
NONE NOSHOW
ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))
curynto = curynto + 10
@ 0,0 SAY " "
@ 0,2 SAY " " COLOR (colorvar)
@ 0,3 SAY " "
@ 0,8 SAY " " COLOR (colorvar)

@ 1,0 SAY " "
@ 1,3 SAY " " COLOR (colorvar)
@ 1,4 SAY " "
@ 1,7 SAY " " COLOR (colorvar)

@ 2,0 SAY " "
@ 2,5 SAY " " COLOR W/H
@ 2,7 SAY " " COLOR (colorvar)

@ 3,3 SAY " " COLOR W/H

@ 4,0 SAY " " COLOR (colorvar)
--CASE SUBSTR(instrng,curlet,1) = "w"
DEFINE WINDOW ("letter" + ALLTRIM(STR(curlet))) ;
FROM startx,curynto startx + 5,curynto + 8 ;
NONE NOSHOW
ACTIVATE WINDOW ("letter" + ALLTRIM(STR(curlet)))

```

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *

```

10/07/92	CAND.WPR	11:51:56
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENMENU.		

```

37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *

```

Menu Definition		
-----------------	--	--

```

37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *

```

```

66 *
67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *

```

```

DEFINE POPUP _cfile MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF _cfile PROMPT "\<Reindex";
SKIP FOR NOT_EMPTY(MONITOR())
DEFINE BAR 2 OF _cfile PROMPT "\<Quit"
ON SELECTION BAR 1 OF _cfile DO REINDEX
ON SELECTION BAR 3 OF _cfile DO _quit

DEFINE POPUP _credit MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _med_undo OF _credit PROMPT "\<Undo";
KEY ctrl+u, "u"
DEFINE BAR _med_redo OF _credit PROMPT "\<Redo";
KEY ctrl+r, "r"
DEFINE BAR _med_sp100 OF _credit PROMPT "\<";
DEFINE BAR _med_cut OF _credit PROMPT "\<Cut";
KEY ctrl+x, "x"
DEFINE BAR _med_copy OF _credit PROMPT "\<Copy";
KEY ctrl+c, "c"
DEFINE BAR _med_paste OF _credit PROMPT "\<Paste";
KEY ctrl+v, "v"
DEFINE BAR _med_clear OF _credit PROMPT "\<Clear"
DEFINE BAR _med_sp200 OF _credit PROMPT "\<";
DEFINE BAR _med_selecta OF _credit PROMPT "\<Select \<All";
KEY ctrl+a, "a"
DEFINE BAR _med_sp300 OF _credit PROMPT "\<";
DEFINE BAR _med_goto OF _credit PROMPT "\<GoTo \<Line...";
DEFINE BAR _med_find OF _credit PROMPT "\<Find...";
KEY ctrl+f, "f"
DEFINE BAR _med_finda OF _credit PROMPT "\<Find A\<gain";
KEY ctrl+g, "g"
DEFINE BAR _med_repl OF _credit PROMPT "\<Replace And Find Again";
KEY ctrl+e, "e"
DEFINE BAR _med_repla OF _credit PROMPT "\<Replace All"
DEFINE BAR _med_sp400 OF _credit PROMPT "\<";
DEFINE BAR _med_pref OF _credit PROMPT "\<Prefer\<nces...";

DEFINE POPUP diagnosis MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF diagnosis PROMPT "\<Register Patient...";
SKIP FOR enc open AND mncstate != "n"
DEFINE BAR 2 OF diagnosis PROMPT "\<Encounter...";
DEFINE BAR 3 OF diagnosis PROMPT "\<SF600 Report"
DEFINE BAR 4 OF diagnosis PROMPT "\<";
DEFINE BAR 5 OF diagnosis PROMPT "\<Disease/Treatment Lookup"
DEFINE BAR 6 OF diagnosis PROMPT "\<";
DEFINE BAR 7 OF diagnosis PROMPT "\<Encounter \<Training Mode"
ON SELECTION BAR 1 OF diagnosis DO _whit WITH "m pat"
ON SELECTION BAR 2 OF diagnosis DO _whit WITH "m newenc"
ON SELECTION BAR 3 OF diagnosis DO _whit WITH "m sf600"
ON SELECTION BAR 5 OF diagnosis DO _whit WITH "m lookup"
ON SELECTION BAR 7 OF diagnosis ;
DO _cddpfic8;
IN LOCFILE("CAND", "WPR;WPR|FX;PRG", "Where is CAND?")

DEFINE POPUP _cwindow MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mwi_move OF _cwindow PROMPT "\<Move";
KEY ctrl+f7, "f7"
DEFINE BAR _mwi_size OF _cwindow PROMPT "\<Size";
KEY ctrl+f8, "f8"
DEFINE BAR _mwi_zoom OF _cwindow PROMPT "\<Zoom 1";
KEY ctrl+f10, "f10"
DEFINE BAR _mwi_min OF _cwindow PROMPT "\<Zoom 1";
KEY ctrl+f9, "f9"
DEFINE BAR _mwi_rotate OF _cwindow PROMPT "\<Cycle";
KEY ctrl+f11, "f11"
DEFINE BAR 6 OF _cwindow PROMPT "\<Color...";

```

```

132 DEFINE BAR _mii_sp200 OF _cwindow PROMPT "\- "
133 DEFINE BAR _mii_cnd OF _cwindow PROMPT "Command" ;
134 KEY ctrl+f2, "f2"
135 DEFINE BAR _mii_debug OF _cwindow PROMPT "\<Debug" ;
136 SKIP FOR lis_debug
137 DEFINE BAR _mii_trace OF _cwindow PROMPT "\<Trace" ;
138 SKIP FOR lis_trace
139 DEFINE BAR _mii_view OF _cwindow PROMPT "\<View"
140 ON SELECTION BAR 6 OF _cwindow DO colorset.spr
141
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *
181 *
182 *
183 *
184 *
185 *
186 *
187 *
188 *
189 *
190 *
191 *
192 *
193 *
194 *
195 *
196 *
197

```

Procedure Origin:

From Menu: CAMD.MPR, Record: 47

Called By: ON SELECTION BAR 7 OF POPUP diagnosis

Prompt: Encounter Training Mode

Snippet: 2

```

198 _qppopflr8 ON SELECTION BAR 7 OF POPUP diagnosis
199
200 *
201 *
202 *
203 *
204 *
205 *
206 *
207 *
208 *
209 *
210 *
211 *
212 *
213 *
214 *
215 *
216 *
217 *
218 *
219 *
220 *
221 *

```

Procedure Origin:

From Menu: CAMD.MPR, Record: 47

Called By: ON SELECTION BAR 7 OF POPUP diagnosis

Prompt: Encounter Training Mode

Snippet: 2

```

222 *
223 *
224 *
225 *
226 *
227 *
228 *
229 *
230 *
231 *
232 *
233 *
234 *
235 *
236 *
237 *
238 *
239 *
240 *
241 *
242 *
243 *
244 *
245 *
246 *
247 *
248 *
249 *
250 *
251 *
252 *
253 *
254 *
255 *
256 *
257 *
258 *
259 *
260 *
261 *
262 *
263 *
264 *
265 *
266 *
267 *
268 *
269 *
270 *
271 *
272 *
273 *
274 *
275 *
276 *
277 *
278 *
279 *
280 *
281 *
282 *
283 *
284 *
285 *
286 *
287 *
288 *
289 *
290 *
291 *
292 *
293 *
294 *
295 *
296 *
297 *
298 *
299 *
300 *
301 *
302 *
303 *
304 *
305 *
306 *
307 *
308 *
309 *
310 *
311 *
312 *
313 *
314 *
315 *
316 *
317 *
318 *
319 *
320 *
321 *
322 *
323 *
324 *
325 *
326 *
327 *
328 *
329 *
330 *
331 *
332 *
333 *
334 *
335 *
336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *

```

Cleanup Code & Procedures

```

PROCEDURE mhit
PARAMETER prog
*
* If an application READ is already active,
* execute CLEAR READ. This causes the active
* application READ to terminate as soon as
* the menu processing routine is complete and
* also requests that the foundation READ be terminated.
* The active application READ terminates first
* and control passes back to the foundation READ.
* Since it has also been CLEARED, it executes
* its VALID routine. This VALID routine, noting
* that the PUBLIC variable 'tobedone' has been
* assigned a value, launches the new application
* which has been requested.

```

```

IF RDLEVEL() > 1
tobedone = PROG
CLEAR READ
ELSE
*
* If no other application is running,
* just start the requested application
DO (PROG)
ENDIF

```

```

PROCEDURE mexit
droptdead = .T.
CLEAR READ ALL

```

Procedure Origin:

From Menu: CAMD.MPR, Record: 7

Called By: ON SELECTION BAR 3 OF POPUP _csystem

Prompt: Change Password

Snippet: 1

```

PROCEDURE _qppopfl0y

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

```

09/28/92	CANDTR.MPR	14:22:23
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENMENU.		

Menu Definition

```

SET SYSTEM TO
SET SYSTEM AUTOMATIC
DEFINE PAD _q9out1ky OF _msysmenu PROMPT "\<System" COLOR SCHEME 3 ;
KEY alt+s ""
DEFINE PAD _q9out1lc OF _msysmenu PROMPT "\<File" COLOR SCHEME 3 ;
KEY alt+f ""
DEFINE PAD _q9out1m7 OF _msysmenu PROMPT "\<Edit" COLOR SCHEME 3 ;
KEY alt+e ""
DEFINE PAD _q9out1mt OF _msysmenu PROMPT "\<Diagnosis" COLOR SCHEME
=> 3 ;
KEY alt+d ""
DEFINE PAD _q9out1nc OF _msysmenu PROMPT "\<Window" COLOR SCHEME 3 ;
KEY alt+n ""
DEFINE PAD _q9out1oc OF _msysmenu PROMPT "\<TRAINING MODE" COLOR SCHE
=> ME 3
ON PAD _q9out1ky OF _msysmenu ACTIVATE POPUP _csystem
ON PAD _q9out1lc OF _msysmenu ACTIVATE POPUP _cfile
ON PAD _q9out1m7 OF _msysmenu ACTIVATE POPUP _ccedit
ON PAD _q9out1mt OF _msysmenu ACTIVATE POPUP _cdiagnosis
ON PAD _q9out1nc OF _msysmenu ACTIVATE POPUP _cwindow
DEFINE POPUP _csystem MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mst help OF _csystem PROMPT "\<Help...";
KEY f1 "F1"
DEFINE BAR 2 OF _csystem PROMPT "\<";
DEFINE BAR 3 OF _csystem PROMPT "Change \<Password";
SKIP FOR NOT EMPTY(WONTOP());
DEFINE BAR 4 OF _csystem PROMPT "\<Backup";
SKIP FOR NOT EMPTY(WONTOP());
DEFINE BAR 5 OF _csystem PROMPT "\<Restore";
SKIP FOR NOT EMPTY(WONTOP());
DEFINE BAR 6 OF _csystem PROMPT "\<";
DEFINE BAR _mst filler OF _csystem PROMPT "\<Filler";
DEFINE BAR _mst calcu OF _csystem PROMPT "\<Calculator";
DEFINE BAR _mst diary OF _csystem PROMPT "\<Calendar\<Diary";
DEFINE BAR _mst puzzle OF _csystem PROMPT "\<Puzzle";
ON SELECTION BAR 3 OF _csystem DO _mst help;
ON SELECTION BAR 4 OF _csystem DO _mst backup;
ON SELECTION BAR 5 OF _csystem DO _mst restore;

```

```

65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130

```

```

DEFINE POPUP _cfile MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF _cfile PROMPT "\<Reindex";
SKIP FOR NOT EMPTY(WONTOP());
DEFINE BAR 2 OF _cfile PROMPT "\<";
DEFINE BAR 3 OF _cfile PROMPT "\<Quit";
ON SELECTION BAR 1 OF _cfile DO REINDEX
ON SELECTION BAR 3 OF _cfile DO _quit
DEFINE POPUP _ccedit MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _med undo OF _ccedit PROMPT "\<Undo";
KEY ctrl+u, "u"
DEFINE BAR _med redo OF _ccedit PROMPT "\<Redo";
KEY ctrl+r, "r"
DEFINE BAR _med sp100 OF _ccedit PROMPT "\<";
DEFINE BAR _med cut OF _ccedit PROMPT "\<Cut";
KEY ctrl+x, "x"
DEFINE BAR _med copy OF _ccedit PROMPT "\<Copy";
KEY ctrl+c, "c"
DEFINE BAR _med paste OF _ccedit PROMPT "\<Paste";
KEY ctrl+v, "v"
DEFINE BAR _med clear OF _ccedit PROMPT "\<Clear";
DEFINE BAR _med sp200 OF _ccedit PROMPT "\<";
DEFINE BAR _med selecta OF _ccedit PROMPT "\<Select \<All";
KEY ctrl+a, "a"
DEFINE BAR _med sp300 OF _ccedit PROMPT "\<";
DEFINE BAR _med goto OF _ccedit PROMPT "\<Goto \<Line...";
DEFINE BAR _med find OF _ccedit PROMPT "\<Find...";
KEY ctrl+f, "f"
DEFINE BAR _med finda OF _ccedit PROMPT "\<Find A\<gain";
KEY ctrl+g, "g"
DEFINE BAR _med repl OF _ccedit PROMPT "\<Replace And Find Again";
KEY ctrl+e, "e"
DEFINE BAR _med repla OF _ccedit PROMPT "\<Replace All";
DEFINE BAR _med sp400 OF _ccedit PROMPT "\<";
DEFINE BAR _med pref OF _ccedit PROMPT "\<Prefer\<nces...";
DEFINE POPUP _cdiagnosis MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR 1 OF _cdiagnosis PROMPT "\<Register Patient...";
SKIP FOR NOT EMPTY(WONTOP());
DEFINE BAR 2 OF _cdiagnosis PROMPT "\<Encounter...";
DEFINE BAR 3 OF _cdiagnosis PROMPT "\<SF600 Report";
DEFINE BAR 4 OF _cdiagnosis PROMPT "\<";
DEFINE BAR 5 OF _cdiagnosis PROMPT "\<Disease/Treatm...< Lookup";
DEFINE BAR 6 OF _cdiagnosis PROMPT "\<";
ON SELECTION BAR 1 OF _cdiagnosis DO _mst pat;
ON SELECTION BAR 2 OF _cdiagnosis DO _mst newenc;
ON SELECTION BAR 3 OF _cdiagnosis DO _mst sf600;
ON SELECTION BAR 5 OF _cdiagnosis DO _mst lookup;
ON SELECTION BAR 7 OF _cdiagnosis DO _mst training;
DEFINE POPUP _cwindow MARGIN RELATIVE SHADOW COLOR SCHEME 4
DEFINE BAR _mst move OF _cwindow PROMPT "\<Move";
KEY ctrl+f7, "f7"
DEFINE BAR _mst size OF _cwindow PROMPT "\<Size";
KEY ctrl+f8, "f8"
DEFINE BAR _mst zoom OF _cwindow PROMPT "\<Zoom 1";
KEY ctrl+f10, "f10"
DEFINE BAR _mst min OF _cwindow PROMPT "\<Zoom 1";
KEY ctrl+f9, "f9"
DEFINE BAR _mst rotat OF _cwindow PROMPT "\<Cycle";
KEY ctrl+ff, "ff"
DEFINE BAR _mst sp200 OF _cwindow PROMPT "\<Color...";
DEFINE BAR _mst chd OF _cwindow PROMPT "\<Command";

```

```

131 KEY Ctrl+F2, "F2="
132 DEFINE BAR _mni_debug OF _cwindow PROMPT "\<Debug" ;
133 SKIP FOR its_debug
134 DEFINE BAR _mni_trace OF _cwindow PROMPT "\<Trace" ;
135 SKIP FOR its_debug
136 DEFINE BAR _mni_view OF _cwindow PROMPT "\<View"
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179

```

Cleanup Code & Procedures

```

PROCEDURE wht
PARAMETER PROG
*
* If an application READ is already active
* execute CLEAR READ. This causes the active
* application READ to terminate as soon as
* the menu processing routine is complete and
* also requests that the foundation READ be terminated.
* The active application READ terminates first
* and control passes back to the foundation READ.
* Since it has also been CLEARED, it executes
* its VALID routine. This VALID routine, noting
* that the PUBLIC variable 'tobedone' has been
* assigned a value, launches the new application
* which has been requested.
*
IF RDLEVEL() > 1
tobedone = PROG
CLEAR READ
ELSE
*
* If no other application is running,
* just start the requested application
*
DO (PROG)
ENDIF

PROCEDURE mexit
droptdead = .T.
CLEAR READ ALL
*.: EOF: CAMDTR.AC2

```

10/07/92	COLORSET.SPR	11:52:12
Author's Name		
Copyright (c) 1992 Company Name		
Address		
City, Zip		
Description:		
This program was automatically generated by GENSCRN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

REGIONAL m.curraree, m.talkstat, m.compatstat
REGION 0
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compatstat = SET("COMPATIBLE")
SET COMPATIBLE FORPLUS

```

Window definitions

```

IF NOT EXIST("w_color")
DEFINE WINDOW w_color
FROM INT((SROW()-14)/2),INT((SCOL()-32)/2)
TO INT((SROW()-14)/2)+13,INT((SCOL()-32)/2)+31;
NOFLOAT;
NOCLOSE;
SHADOW;
DOUBLE;
COLOR SCHEME 5
ENDIF

```

COLORSET Setup Code - SECTION 2

```

REGION 1
IF USED("colorset")
SELECT 0
USE colorset INDEX colorset
ELSE
SELECT colorset
ENDIF
COPY TO ARRAY colorlist FIELD color_set

```

COLORSET Screen Layout

```

REGION 1
IF WVISIBLE("w_color")
ACTIVATE WINDOW w_color SAME
ELSE
ACTIVATE WINDOW w_color NOSHOW
ENDIF
@ 0,1 GET m.coloridx;
PICTURE "a.m";
FROM colorlist;
SIZE 10,27;
DEFAULT 1;
COLOR SCHEME 6
@ 11,5 GET m.action;
PICTURE "a.m";
SIZE 1,8,3;
DEFAULT 1;
VALID _qppopfa2h()

```

```

IF NOT WVISIBLE("w_color")
ACTIVATE WINDOW w_color
ENDIF

```

READ CYCLE

RELEASE WINDOW w_color

```

REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compatstat = "ON"
SET COMPATIBLE ON
ENDIF

```

Function Origin:

From Screen: COLORSET, Record Number: 3

Variable: m.action

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 1

```

FUNCTION _qppopfa2h
REGION 1
IF m.action = 1
m.colorset = TRIM(colorlist(m.coloridx))
IF m.colorset = "COLOR"
m.colorset = "DEFAULT"
ENDIF
SET COLOR SET TO (m.colorset)
ENDIF
*; EOF: COLORSET.AC1

```



```

133 * Medical Practice Support System
134 * Revision: 1.1 $ $Date: 92/05/04 11:10:24 $
135 * Author: RoyDobbins $
136 * 4/21/92 rwd01 popup warning when assist run without all
137 * symptoms entered by user
138 * 4/27/92 rwd03 batch test mode
139 * 6/1/92 rwd05 handle empty ruleset
140 * 6/2/92 rwd07 performance tuning of assist
141 * 6/9/92 hl101 delete variable "startdir"
142 * 6/10/92 hl102 add procedures and functions for patient integrate
=> with SAMS
143 * 6/10/92 hl103 add procedures and functions for doing sf600 report
144 * 6/18/92 hl104 update patient info (sex and age) to working memory
145 * 6/18/92 hl105 Comment the computer diagnosis out from sf600 report
146 * 6/18/92 hl106 add a report to PUBLIC variable for getting the sf600
=> report
147
148 PARAMETERS testdir, testfile, logfile      ## rwd03
149
150 * These parameters are only required for testing.      ## rwd03
151 * By default, the system runs in interactive mode      ## rwd03
152 * and none of these parameters are then required:      ## rwd03
153
154 * testdir is the name of the test directory      ## rwd03
155 * This directory is the location of the patient      ## rwd03
156 * encounter transaction files used for testing.      ## rwd03
157
158 * testfile is the name of the test driver file      ## rwd03
159 * This file lists the set of test cases to be run      ## rwd03
160
161 * logfile is the name of the log file      ## rwd03
162 * This file records a summary of the test results      ## rwd03
163 PUBLIC talk, wproc
164 IF SET("TALK") = "ON"
165 SET TALK OFF
166 talk = SET("TALK")
167
168 userValid = secheck()
169 IF userValid
170 RETURN
171
172 ENDIF
173 =userInit()
174
175 * delay expected here ...
176 PRIVATE wpopup
177 wpopup = popupshou("initializing...")
178 =inEncounter()
179 =inSymptoms()
180 =inDiagnoses()
181 =popupHide(wpopup)
182
183
184
185
186
187
188 * main application menu
189
190 * system state vector for knowledge base processing
191 PRIVATE statevecf
192 PRIVATE statevecq
193 PRIVATE fire
194 PUBLIC agenda
195 PRIVATE ruleaction
196 * to be re-dimensionalized
197 DIMENSION statevecf(10), statevecq(10), fire(10), ruleaction(10) ##

```

```

196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

260 DIMENSION mqld(alen(mcondition))
261
262 event = e_ok
263
264 * set up working storage for the encounter screen
265 mdiagname = ""
266 mcond = ""
267 mqld = ""
268 nc = strcond(second, mqld, mcondition, maysymval)
269 n = strdiag(mdiagname, mdiag, mconf, cthreshold, "", is_conf)
270 marea = getxarea(marea)
271 mprdiagname = disasname(mdiagname)
272 mstatename = mncstate
273
274 * edit the screen
275 mge = ageat(mdateenc, mdatebirth)
276 msex = sexage(mage, msex)
277 mevent = encntitle(mocsec, mlastname, mfirstname, msex, mage,
278 mdateenc, mtimeenc, mareaenc, mprovider,
279 mprdiagname, mdiagname, mcond, mtrtreatment, mstatename)
280
281 * close the screen
282
283 * respond to one of the following pushbutton events
284
285 -DO CASE
286   CASE mevent = e_save
287     * save encounter
288     * save encounter
289     * save encounter
290     * save encounter
291     * save encounter
292     * save encounter
293     * save encounter
294     * save encounter
295     * save encounter
296     * save encounter
297     * save encounter
298     * save encounter
299     * save encounter
300     * save encounter
301     * save encounter
302     * save encounter
303     * save encounter
304     * save encounter
305     * save encounter
306     * save encounter
307     * save encounter
308     * save encounter
309     * save encounter
310     * save encounter
311     * save encounter
312     * save encounter
313     * save encounter
314     * save encounter
315     * save encounter
316     * save encounter
317     * save encounter
318     * save encounter
319     * save encounter
320     * save encounter
321     * save encounter
322     * save encounter
323     * save encounter
324     * save encounter
325     * save encounter
326     * save encounter
327     * save encounter
328     * save encounter
329     * save encounter
330     * save encounter
331     * save encounter
332     * save encounter
333     * save encounter
334     * save encounter
335     * save encounter

```

```

336 *
337 *
338 *
339 *
340 *
341 *
342 *
343 *
344 *
345 *
346 *
347 *
348 *
349 *
350 *
351 *
352 *
353 *
354 *
355 *
356 *
357 *
358 *
359 *
360 *
361 *
362 *
363 *
364 *
365 *
366 *
367 *
368 *
369 *
370 *
371 *
372 *
373 *
374 *
375 *
376 *
377 *
378 *
379 *
380 *
381 *
382 *
383 *
384 *
385 *
386 *
387 *
388 *
389 *
390 *
391 *
392 *
393 *
394 *
395 *
396 *
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
415 *
416 *
417 *
418 *
419 *
420 *
421 *
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 *
438 *
439 *
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *
460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *
525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 *
541 *
542 *
543 *
544 *
545 *
546 *
547 *
548 *
549 *
550 *
551 *
552 *
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 *
569 *
570 *
571 *
572 *
573 *
574 *
575 *
576 *
577 *
578 *
579 *
580 *
581 *
582 *
583 *
584 *
585 *
586 *
587 *
588 *
589 *
590 *
591 *
592 *
593 *
594 *
595 *
596 *
597 *
598 *
599 *
600 *
601 *
602 *
603 *
604 *
605 *
606 *
607 *
608 *
609 *
610 *
611 *
612 *
613 *
614 *
615 *
616 *
617 *
618 *
619 *
620 *
621 *
622 *
623 *
624 *
625 *
626 *
627 *
628 *
629 *
630 *
631 *
632 *
633 *
634 *
635 *
636 *
637 *
638 *
639 *
640 *
641 *
642 *
643 *
644 *
645 *
646 *
647 *
648 *
649 *
650 *
651 *
652 *
653 *
654 *
655 *
656 *
657 *
658 *
659 *
660 *
661 *
662 *
663 *
664 *
665 *
666 *
667 *
668 *
669 *
670 *
671 *
672 *
673 *
674 *
675 *
676 *
677 *
678 *
679 *
680 *
681 *
682 *
683 *
684 *
685 *
686 *
687 *
688 *
689 *
690 *
691 *
692 *
693 *
694 *
695 *
696 *
697 *
698 *
699 *
700 *
701 *
702 *
703 *
704 *
705 *
706 *
707 *
708 *
709 *
710 *
711 *
712 *
713 *
714 *
715 *
716 *
717 *
718 *
719 *
720 *
721 *
722 *
723 *
724 *
725 *
726 *
727 *
728 *
729 *
730 *
731 *
732 *
733 *
734 *
735 *
736 *
737 *
738 *
739 *
740 *
741 *
742 *
743 *
744 *
745 *
746 *
747 *
748 *
749 *
750 *
751 *
752 *
753 *
754 *
755 *
756 *
757 *
758 *
759 *
760 *
761 *
762 *
763 *
764 *
765 *
766 *
767 *
768 *
769 *
770 *
771 *
772 *
773 *
774 *
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *

```



```

117 *ELSE
118   =popmessage("Insufficient data for diagnosis")
119   =resetdiag()
120 ENDIF
121 *ELSE
122   =popuhide(wpopup)
123   =popmessage("Complaint must be entered first")
124   =RELEASE("% assist")
125   CUIROGJ = OBJNUM(mickcomp1)
126   SHOW GETS
127 ENDIF
128 RETURN
129 *****
130 *
131 * menu service procedure - enter signs and symptoms
132 *****
133 *****
134 *****
135 Procedure: M_HANDIP
136 Called by: _EDITHANDIP      (procedure in U.PRG)
137 Calls: POPUPSHOW()         (function in U.PRG)
138       : TESTRULES          (procedure in DEMO.PRG)
139       : POPUPHIDE()        (function in U.PRG)
140       : POPMESSAGE()       (function in U.PRG)
141 *****
142 *****
143 PROCEDURE m_handip
144 * only allowed after complaint has been entered
145 IF IEMPTY(marea)
146   * gather signs and symptoms data
147   * delay expected here ...
148 PRIVATE wpopup
149 wpopup = popupshow("Checking...")
150 event = e_ok
151 testrules(marea)
152 =popuhide(wpopup)
153 *ELSE
154   =popmessage("Complaint must be entered first")
155   =RELEASE("% handip")
156   CUIROGJ = OBJNUM(mickcomp1)
157   SHOW GETS
158 ENDIF
159 RETURN
160 *****
161 *
162 * menu service procedure - provide treatment recommendation
163 *****
164 *****
165 *****
166 Procedure: M_TREAT
167 Called by: _EDITTREAT      (procedure in U.PRG)
168 Calls: POPMESSAGE()       (function in U.PRG)
169 *****
170 *****
171 PROCEDURE m_treat
172 PARAMETERS ntreat
173 * only allowed after diagnosis has been entered
174 IF IEMPTY(mdifgnosis)

```

```

583 mtreatment = newtreat
584 CUROBJ = OBJNUM(me_treat)
585 ELSE
586     *message("diagnosis must be entered first")
587     *RELEASE("u_treat")
588     CUROBJ = OBJNUM(mckdlog)
589 ENDIF
590 RETURN
591
592 *
593 *
594 * menu service procedure - redo qualifier
595 *
596 *****
597 Procedure: M_REDOQ
598
599 Called by: _EDITCOMD
600
601 Calls: GETQ
602         : ERASEQ
603         : ASK
604         : RESTOREQ
605
606 (procedure in DEMO.PRG)
607 (procedure in DEMO.PRG)
608 (procedure in DEMO.PRG)
609 (procedure in DEMO.PRG)
610
611 *****
612 FUNCTION m_redoq
613 PARAMETERS mitem
614 * This procedure is triggered by the user selecting an item
615 * in the symptom list. Wants to edit the value
616 * mitem is the selected item
617 * get its id from the mqid array stored when the screen is
618 * opened (see PADC.m_resume)
619 PRIVATE mid, xstr, mresult, mprev
620 mresult = .f.
621 event = e.ok
622 IF mitem > 0
623     mid = mqid(mitem)
624     xstr = ""
625     mprev = getq(mid)
626     mresult = ask(mid)
627     IF mresult
628         *restoreq(mid, mprev)
629     ENDIF
630 ENDIF
631 RETURN mresult
632
633 *****
634 *
635 * select patient encounter
636 *
637 *****
638 Procedure: ENCOUNDT
639
640 Called by: M_ENC
641
642 Calls: GETENCNT
643         : PATSHR()
644         : M_PATGET
645         : M_ENCGET
646         : M_ENCNEW
647
648 (procedure in DEMO.PRG)
649 (function in U.PRG)
650 (procedure in DEMO.PRG)
651 (procedure in DEMO.PRG)
652 (procedure in DEMO.PRG)
653
654 *****
655
656 *****
657 *****
658 *****
659 *****
660 *****
661 *****

```

```

649 FUNCTION encounter
650 PRIVATE aprevpat
651 aprevpat = patientid
652 patientid = getencnt(patientid, mdateenc)
653 DO CASE
654 CASE patientid > 0
655     * pending encounter
656     IF patientid = aprevpat
657         * selected different patient
658         mdateenc = patientid
659         * get patient record
660         * patientid, mdateenc, lastname, firstname, mdatebirth
661         * h, sex, dateofbirth
662     ENDIF
663 CASE patientid = 0
664     * new encounter
665     * encnt()
666     * aborted encounter
667     patientid = 0
668     RETURN patientid
669 END CASE
670 RETURN patientid
671
672 *-----
673 *
674 * encounter event procedures
675 *
676 *-----
677
678 *****
679 Procedure: M_ENCLOS
680 Called by: M_ENCLOS (procedure in DEMO.PRG)
681
682 *****
683 FUNCTION m_enclo
684 * close encounter without saving
685 enc_open = .F.
686 mdateenc = .
687 RETURN .T.
688
689 *****
690 Procedure: M_ENCSAV
691 Called by: BATCHTEST()
692           : M_ENCSAV
693           : (function in DEMO.PRG)
694           : (procedure in DEMO.PRG)
695
696 Calls: POPUPSHOW()
697       : LODENC
698       : M_ENCLOS
699       : AGEAT
700       : EDSEAGE
701       : POPUPHIDE()
702
703 *****
704 FUNCTION m_encsav
705 * save encounter
706 IF mdateenc = "C"
707     * message("Encounter is closed")
708     RETURN .F.
709 ENDIF

```

```

714 IF .NOT. ( enc_open AND m_enclo )
715     * message("Encounter is empty, not saved")
716     RETURN .F.
717 ENDIF
718 IF mdateenc = "M"
719     * save new encounter
720     mdateenc = "M"
721     * addencnt()
722     * newencnt(patientid, patientid, mdateenc, mdateenc, ;
723       * provider, mdateenc, mdateenc, mdateenc, mdateenc, ;
724       mdateenc)
725 ELSE
726     * save pending encounter
727     * putencnt(patientid, patientid, mdateenc, mdateenc, ;
728       * provider, mdateenc, mdateenc, mdateenc, mdateenc, ;
729       mdateenc)
730 ENDIF
731 * save symptoms/computer's diagnosis for new/pending encounter
732 * putsymptom(patientid, mdateenc, mdateenc)
733 * putdiagnos(patientid, mdateenc, mdateenc)
734 RETURN .T.
735
736 *****
737 Procedure: M_ENCVALID
738 Called by: M_ENCLOS
739           : M_ENCLOS
740           : M_ENCSAV
741
742 *****
743 FUNCTION m_encvalid
744 * test validity of encounter
745 * encounter must have the following minimum attributes:
746 * IF EMPTY(patientid).OR. EMPTY(mdateenc).OR. EMPTY(mdateenc)
747 RETURN .F.
748 * encounter is missing some vital attributes
749 ENDIF
750 RETURN .T.
751 * encounter is valid
752
753 *****
754 Procedure: M_ENCGET
755 Called by: BATCHTEST()
756           : ENCLOS
757           : (function in DEMO.PRG)
758           : (procedure in DEMO.PRG)
759
760 Calls: POPUPSHOW()
761       : LODENC
762       : M_ENCLOS
763       : AGEAT
764       : EDSEAGE
765       : POPUPHIDE()
766
767 *****
768 FUNCTION m_encget
769 * resume previous encounter
770 * delay expected here ...
771 PRIVATE upopup
772 upopup = popupshow("Loading...")
773 enc_open = .T.
774 * loadenc(patientid)
775 * mdateenc
776 * age
777 * edseage(mage, msex)
778 * popuphide(upopup)

```

```

780 *****
794 Procedure: M_ENCNEW
795
796 Called by: ENCOUNTER
797
798 Calls: NEXTENCNT()
799
800 : AGENT (function in U.PRG)
801 : DELSYMPTOM (procedure in DEMO.PRG)
802 : DELDIAGNOS (procedure in DEMO.PRG)
803 : EDSEXAGE (procedure in DEMO.PRG)
804
805 *****
806 FUNCTION m_encnew
807
808 * start a new blank encounter
809 enc_open = .T.
810
811 mncountid = nextencnt()
812 mdateenc = DATE()
813 mtimeenc = TIME()
814
815 * identifier for this encounter
816 * today's date
817 * current time is start of encounter
818 * empty area
819 * empty provider's diagnosis
820 * empty treatment description
821 * ageat(mdateenc, mdatebirth) ** patient's age in years
822 * clear the list of symptoms/diagnoses for new encounter
823 * delsymptom()
824 * deldiagnos()
825 * edsexage(mage, msax)
826 RETURN .T.
827
828 *****
829 * edit basic patient history details to working memory
830 * edit h1104
831
832 *****
833 Procedure: EDSEXAGE
834
835 Called by: M_RESUME
836 : M_ENCGET (procedure in DEMO.PRG)
837 : M_ENCNEW (procedure in DEMO.PRG)
838
839 Calls: ADDHISTORY
840 (procedure in DEMO.PRG)
841
842 *****
843 FUNCTION edsexage
844
845 PARAMETER mage, msax
846 PRIVATE mstr, mval
847
848 DIMENSION mstr(2), mval(2)
849 mstr(1) = "AGE"
850 mstr(2) = "SEX"
851 mval(1) = mage
852 mval(2) = msax
853
854 addhistory(mstr, mval)
855 RETURN .T.
856
857 *****
858 Procedure: M_ENCNEW
859
860 Called by: BATCHTEST()
861 : M_COMPL (function in DEMO.PRG)
862 : M_ENCGET (procedure in DEMO.PRG)
863
864 *****

```

DEMO.ACT 10-16-92 10:50s

```

857 *****
858 FUNCTION m_encrule
859
860 * load rulebase
861 =lodrule(marea)
862 RETURN .T.
863
864 *****
865 * Expert system inference engine ***
866
867 *****
868 * load a rulebase
869
870 *****
871 Procedure: LODRULE
872
873 Called by: M_ENCNEW (procedure in DEMO.PRG)
874
875 Calls: GETRULEBASE() (function in U.PRG)
876
877 *****
878 FUNCTION lodrule
879
880 PARAMETERS marea
881 rulebase = getrulebase(marea, actthreshld, actthreshpro, actthreshlik,
882 * mrules)
883
884 * If rules > 0
885 * DIMENSION statevecf(mrules), statevecq(mrules), fire(mrules) **
886
887 * DIMENSION fire(mrules) **
888 * DIMENSION ruleaction(mrules) **
889
890 * statevecq = ""
891 * statevecf = ""
892 * fire = .F.
893 * ruleaction = 0
894 * agenda = .F.
895 * rule = 1
896 RETURN .T.
897
898 *****
899 * test a rule set
900
901 *****
902 Procedure: TESTRULES
903
904 Called by: M_SSIST (procedure in DEMO.PRG)
905 : M_HANDOP (procedure in DEMO.PRG)
906
907 Calls: EVALCOMO (procedure in DEMO.PRG)
908
909 *****
910 FUNCTION testrules
911
912 PARAMETERS marea
913
914 PRIVATE mresult
915 PRIVATE msel
916 msel = SELECT()
917
918 * action

```

Page 7 of 37

```

908  IF agenda
909  => 22 test for agenda complete
910  SELECT rule
911  GOTO TOP
912  DO WHILE IEOF()
913  * test rules in salience (priority) order
914  * IF is rule
915  * * display rule for debugging
916  * * viewrule(mrule)
917  * * IF is rule
918  * * IF is rule
919  * * IF is rule
920  SELECT premise
921  * test premise clauses of rule
922  * logical AND of all clauses
923  mresult = .T.
924  * result of clause evalu
925  SEEK rule->premise
926  DO WHILE clause = rule->premise
927  IF Ievalcond( OBJECT, ALLTRIM( STR(id) ), op, VAL ) 22
928  mresult = .F.
929  EXIT
930  ENDIF
931  SKIP
932  ENDDO
933  IF event = e_esc
934  * abort flag has been set
935  EXIT
936  ELSE
937  fire(mrule) = mresult
938  ruleaction(mrule) = rule->action
939  22 add rule to agenda
940  22 keep rule clause
941  ENDIF
942  SELECT rule
943  SKIP
944  ENDDO
945  IF event != e_esc
946  agenda = .T.
947  ENDIF
948  SELECT (mrule)
949  RETURN event
950  *
951  *
952  * evaluate rule agenda
953  *
954  *
955  *
956  *
957  *
958  *
959  *
960  *
961  *
962  *
963  *
964  *
965  *
966  *
967  *
968  *
969  *
970  *
971  *
972  *
973  *
974  *
975  *
976  *
977  *
978  *
979  *
980  *
981  *
982  *
983  *
984  *
985  *
986  *
987  *
988  *
989  *
990  *
991  *
992  *
993  *
994  *
995  *
996  *
997  *
998  *
999  *
1000  *
1001  *
1002  *
1003  *
1004  *
1005  *
1006  *
1007  *
1008  *
1009  *
1010  *
1011  *
1012  *
1013  *
1014  *
1015  *
1016  *
1017  *
1018  *
1019  *
1020  *
1021  *
1022  *
1023  *
1024  *
1025  *
1026  *
1027  *
1028  *
1029  *
1030  *
1031  *
1032  *
1033  *
1034  *
1035  *
1036  *
1037  *
1038  *
1039  *
1040  *

```

```

968  PRIVATE msel
969  msel = SELECT()
970  => action
971  PRIVATE maxrules
972  maxrules = ALEN(fire)
973  *SELECT rule
974  *SET ORDER TO 2
975  SELECT action
976  22 rwd07
977  FOR mrule = 1 TO maxrules
978  IF fire(mrule)
979  *SEEK mrule
980  *SELECT action
981  SEEK ruleaction(mrule)
982  DO WHILE clause = ruleaction(mrule)
983  =evalcond( OBJECT, ALLTRIM( STR(id) ), op, VAL ) 22 rwd
984  SKIP
985  ENDDO
986  *SELECT rule
987  IF event = e_esc
988  EXIT
989  ENDIF
990  NEXT
991  *SELECT RULE
992  *SET ORDER TO 1
993  SELECT (msel)
994  RETURN event
995  22 rwd07
996  22 rwd07
997  *
998  *
999  *
1000  *
1001  *
1002  *
1003  *
1004  *
1005  *
1006  *
1007  *
1008  *
1009  *
1010  *
1011  *
1012  *
1013  *
1014  *
1015  *
1016  *
1017  *
1018  *
1019  *
1020  *
1021  *
1022  *
1023  *
1024  *
1025  *
1026  *
1027  *
1028  *
1029  *
1030  *
1031  *
1032  *
1033  *
1034  *
1035  *
1036  *
1037  *
1038  *
1039  *
1040  *

```



```

1159 *
1160 *-----
1171 *****
1172 *****
1173 *****
1174 *****
1175 *****
1176 *****
1177 *****
1178 *****
1179 *****
1180 *****
1181 *****
1182 *****
1183 *****
1184 *****
1185 *****
1186 *****
1187 *****
1188 *****
1189 *****
1190 *****
1191 *****
1192 *****
1193 *****
1194 *****
1204 *****
1205 *****
1206 *****
1207 *****
1208 *****
1209 *****
1210 *****
1211 *****
1212 *****
1213 *****
1214 *****
1215 *****
1216 *****
1217 *****
1218 *****
1219 *****
1220 *****
1221 *****
1222 *****
1223 *****

Procedure: UNDORULES
Called by: M_UNDORULE (procedure in DEMO.PRG)
Calls: UNDOFACTOR (procedure in DEMO.PRG)
        : UNDOQUAL (procedure in DEMO.PRG)

*****
FUNCTION undorules
* undo actions of rules that have fired
* use the string "fire" to find previously active rules
* fire consists of a string of 'f's and 'i's
* backtrack through all rules
PRIVATE i
i = ALEN(fire)
DO WHILE i > 0
* rule is position in string
IF fire(i)
* this rule was active, undo possible actions
undoqual(statevecf(i))
undoqual(statevecq(i))
fire(i) = .f.
ENDIF
i = i - 1
ENDDO
RETURN .T.
*-----
*
* Undo factors from state vector
*-----
*****
Procedure: UNDOFACTOR
Called by: UNDORULES (procedure in DEMO.PRG)
Calls: FACTORID (procedure in DEMO. "G")

*****
FUNCTION undofactor
PARAMETERS statestr
* this restores the system state to what it was BEFORE the
* rule fired, that is a complete "roll-back"
PRIVATE i, eolpos, arrowpos, mstr, mval
DO WHILE LEN(statestr) > 0
* extract an item of the form: name <arrow> val <eol>
arrowpos = AT(arrow, statestr)
eolpos = AT(eol, statestr)
IF arrowpos = 0
EXIT
ENDIF
* extract left hand side and right hand side
mstr = LEFT(statestr, arrowpos - 1)
mval = SUBSTR(statestr, arrowpos + LEN(arrow), eolpos - (ar
=> rompos + LEN(arrow)))
i = qid(mstr)
IF mval = uval
* undefined value, delete element
asynpval(i) = ""
ELSE
* restore previous value
asynpval(i) = mval
ENDIF
* remove last item processed
statestr = LEFT(statestr, SUBSTR(statestr, SUBSTR(statestr, eolpos +
=> 1), "n"))
ENDDO
RETURN .T.
*-----
*
* ask providers diagnosis
*-----
*****

```



```

1297 *****
1299 Procedure: ASKDIAG
1300
1301 Called by: M_DIAG (procedure in DEMO.PRG)
1302
1303 Calls: DISEASECNT() (function in U.PRG)
1304 : DISEASECNT() (function in U.PRG)
1305 : GETDIAGLIST() (function in U.PRG)
1306
1307 *****
1308 FUNCTION askdiag
1309 PARAMETERS mdiagnois
1310
1311 PRIVATE names, M, ndis, mchoice, i
1312 M = diseasecnt()
1313 mchoice = ""
1314 IF M > 0
1315 DIMENSION names(n)
1316 ndis = dnames(names, mdiagnois)
1317
1318 i = 1
1319 IF i > 0, getdiaglist(names), 0 )
1320 IF i > 0
1321 * extract disease from chosen entry
1322 mchoice = names(i)
1323 ENDIF
1324 RETURN mchoice
1325
1326 *****
1327 * select diagnostic area
1328
1329 *****
1330 Procedure: GETDIAGAREA
1331 Called by: M_CMPL (procedure in DEMO.PRG)
1332 : M_DATA (procedure in DEMO.PRG)
1333
1334 Calls: AREACOUNT() (function in U.PRG)
1335 : AREANAMES() (function in U.PRG)
1336 : GETAREALIST() (function in U.PRG)
1337
1338 *****
1339 FUNCTION getdiagarea
1340
1341 * get list of diagnostic area choices
1342 PRIVATE names, M
1343 M = areacount()
1344 DIMENSION areas(n), names(n)
1345 areanames(areas, names)
1346 i = getarealist(names)
1347
1348 RETURN IF(i > 0, areas(i), 0)
1349
1350 *****
1351 * get patient from file
1352
1353 *****
1354 PRIVATE M
1355 PRIVATE names
1356 PRIVATE areaidid
1357
1358 *****

```

```

1362 *****
1363 Procedure: GETPATIENT
1364 Called by: M_PATLIST (procedure in DEMO.PRG)
1365
1366 Calls: PATCOUNT() (function in U.PRG)
1367 : PATNAMES() (function in U.PRG)
1368 : GETPATLIST() (function in U.PRG)
1369
1370 *****
1371 FUNCTION getpatient
1372 PARAMETERS mkey
1373
1374 * mkey is the search key:
1375 * if it is string numeric => social security number
1376 * otherwise => lastname
1377
1378 *****
1379 PRIVATE DELIM
1380 PRIVATE names, M, rpts, mchoice, i
1381 M = patcount()
1382 IF M > 0
1383 DIMENSION names(n)
1384 DELIM = ""
1385 rpts = patnames(names, DELIM, mkey)
1386
1387 * Get users choice from menu of names:
1388 * > 0 selected patient esn
1389 * -1 user aborted
1390 i = 1
1391 IF rpts > 0, getpatlist(mkey, names), 0)
1392 IF i > 0
1393 * extract socsec number from chosen entry
1394 mchoice = SUBSTR(names(i), 1, AT(DELIM, names(i)) - 1)
1395 IF mchoice = ""
1396 mchoice = ""
1397 ENDIF
1398 ELSE
1399 mchoice = ""
1400 ENDIF
1401 RETURN mchoice
1402
1403 *****
1404 * get previous patient encounter for processing/review
1405
1406 *****
1407 Procedure: GETENCNT
1408 Called by: ENCOUN (procedure in DEMO.PRG)
1409
1410 Calls: ENCOUNCNT() (function in U.PRG)
1411 : ENCOUNCNT() (function in U.PRG)
1412 : GETENCLIST() (function in U.PRG)
1413 : PATSID() (function in U.PRG)
1414
1415 *****
1416 FUNCTION getencnt
1417 PARAMETERS espatientid, mdate
1418
1419 PRIVATE M
1420 PRIVATE names
1421 PRIVATE areaidid
1422
1423 *****

```

```

1416 mncntid = 0
1417 PRIVATE names, ids, patients, M, mchoic
1418 M = encounter()
1419 IF M > 0
1420     M = number of encounters
1421     user selects an encounter from list
1422     DIMENSION names(n), ids(n), patients(n)
1423     M = encnames(names, ids, patients, mpatientid, mdate)
1424     create list of names
1425     IF M > 0
1426         DIMENSION names(n), ids(n), patients(n)
1427         mchoic = getenclist(names, IF(TYPE("mpatientid") = "M", pe
1428             IF mchoic > 0
1429                 mncntid = ids[mchoic]
1430                 mpatientid = patients[mchoic]
1431             ELSE
1432                 mncntid = mchoic
1433             ENDIF
1434         ENDIF
1435     RETURN mncntid
1436
1437 *
1438 * initialize encounter details
1439 *
1440
1441 *****
1442 Function: INTENCOUNT()
1443 *****
1444 Called by: DEMO.PRG
1445
1446 Calls: GETSTATION() (function in U.PRG)
1447 *****
1448 FUNCTION intencount
1449
1450 * global variables describing the current encounter
1451
1452 * details about the encounter
1453 PUBLIC mncntid
1454
1455 * unique identifier for the encounter
1456
1457 PUBLIC mdateenc
1458 PUBLIC mtimeenc
1459
1460 * date of the encounter
1461 * time of the encounter (starting t
1462
1463 PUBLIC mstation
1464
1465 * name of duty station for the enco
1466
1467 PUBLIC mncntate
1468
1469 * encounter state
1470 * encounter sf600 report
1471
1472 * details about the patient
1473 PUBLIC mpatientid
1474 PUBLIC mncntsec
1475 PUBLIC mlastname
1476 PUBLIC mfirstname
1477 PUBLIC mmiddleinitial
1478 PUBLIC mdatebirth
1479 PUBLIC msex
1480 PUBLIC mrank
1481 PUBLIC mdept
1482 PUBLIC mage
1483
1484 * details about the diagnosis
1485
1486 *****
1487 PUBLIC marea
1488 PUBLIC mdiagnosis
1489 PUBLIC mcomplaint
1490 PUBLIC mtrmt
1491 PUBLIC mstate
1492
1493 * initialize a null encounter record
1494 mncntid = 0
1495 mpatientid = 0
1496 mlastname = ""
1497 mfirstname = ""
1498 mmiddleinitial = ""
1499 mdatebirth = CDATE(" / / ")
1500 msex = ""
1501 mrank = ""
1502 mdateenc = DATE()
1503 mtimeenc = TIME()
1504 marea = 0
1505 mdiagnosis = 0
1506 mcomplaint = ""
1507 mtrmt = ""
1508 mstate = ""
1509
1510 * get station details
1511 *getstation(mstation, mdept)
1512
1513 RETURN .T.
1514
1515 * load patient encounter record
1516 *****
1517
1518 Procedure: LODEMC
1519
1520 Called by: M ENCGET
1521 (procedure in DEMO.PRG)
1522 (procedure in U.PRG)
1523
1524 Cells: ENCGETINFO()
1525 : DELDIAGNOSIS
1526 : GETDIAGNOSIS()
1527 : GETSYMPTOM
1528 : GETSYMPTOM()
1529 (function in U.PRG)
1530 (function in DEMO.PRG)
1531 (function in U.PRG)
1532 (function in DEMO.PRG)
1533 (function in U.PRG)
1534 (function in DEMO.PRG)
1535 (function in U.PRG)
1536
1537 *****
1538 FUNCTION lodeenc
1539
1540 PARAMETERS mncntid
1541
1542 * get details from (selected) encounter database record
1543 *getinfo(mncntid, mpatientid, mdateenc, mtimeenc, mprovider, ;
1544     mstation, mdept, marea, mdiagnosis, mtrmt, mncntate, mst
1545
1546 * load list of computer's diagnoses for this encounter
1547 *loaddiagnos()
1548 *load current diagnoses
1549 *getdiagnos(mncntid)
1550 *load the list of symptoms for this encounter
1551 *loadsymptom()
1552 *getsymptom(mncntid)
1553
1554 *****

```

```

1539 RETURN mcountid
1540
1541
1542 * Diagnostic assistance
1543 *
1544
1545
1546
1547
1548 Procedure: M_CHANGE
1549
1550 Calls: CHANGEQ (procedure in DEMO.PRG)
1551 : POPQ (procedure in DEMO.PRG)
1552 : M_UNDORULE (procedure in DEMO.PRG)
1553 : POPMESSAGE() (function in U.PRG)
1554
1555
1556
1557 FUNCTION m_change
1558 event = e_redo  ** redo current question
1559 IF equal > 1
1560 * change data, only if stack not empty
1561 * present choice of data to change
1562 mq = changeq(mqual-1)
1563 IF mq > 0
1564 * user made a selection
1565 *popq(mq) ** pop questions from stack
1566 event = e_goback
1567 ** m_undorule()
1568 ELSE
1569 *popmessage("This is first question")
1570 ENDIF
1571 RETURN .T.
1572
1573
1574 Procedure: M_UNDO
1575
1576 Calls: LASTQ (procedure in DEMO.PRG)
1577 : POPQ (procedure in DEMO.PRG)
1578 : M_UNDORULE (procedure in DEMO.PRG)
1579
1580
1581 FUNCTION m_undo
1582 * go back (remove most recent question from stack)
1583 mq = lastq()
1584 *popq(mq) ** index of last question
1585 ** pop previous question from stack
1586 ** m_undorule()
1587 RETURN .T.
1588
1589
1590 Procedure: M_UNDORULE
1591
1592 Called by: M_CHANGE (procedure in DEMO.PRG)
1593 : M_UNDO (procedure in DEMO.PRG)
1594
1595
1596 Calls: UNDORULES (procedure in DEMO.PRG)
1597
1598
1599 FUNCTION m_undorule
1600 *undorules()
1601 ** restore data from rule state vector
1602 ** reset state of qualifiers
1603 ** state of confidence factors
1604 ** reset rules fired
1605 ** begin at first rule
1606 mrule = 1
1607

```

DEMO.ACT 10-14-92 10:50a

```

1605 RETURN .T.
1606
1607
1608 Procedure: MYUNDO
1609
1610 Calls: UNDOMENU() (function in U.PRG)
1611
1612
1613
1614 FUNCTION myundo
1615 IF event = e_esc
1616 * user hit abort key during data entry
1617 ** popup menu
1618 event = undomenu()
1619 ENDIF
1620 RETURN event
1621
1622
1623
1624 * make a diagnosis
1625
1626
1627
1628 Procedure: MAKEDIAG
1629
1630 Called by: M_ASSIST (procedure in DEMO.PRG)
1631
1632
1633 Calls: NORMFACTOR (procedure in DEMO.PRG)
1634
1635
1636 FUNCTION makediag
1637 * NOTE!
1638 * This version does only Bayesian (multiplicative)
1639 * confidence factors. An enhanced version should do more
1640 * (eg fuzzy, additive, etc). The method of
1641 * computation should be set as a parameter:
1642 * I DO CASE
1643 * I CASE cfmethod = "BAYES"
1644 * normalize probabilities already calculated
1645 *normfactor()
1646 * I CASE cfmethod = "ADDITIVE"
1647 * I CASE cfmethod = "FUZZY"
1648 * I ENDCASE
1649 RETURN .T.
1650
1651
1652
1653 * get current patient info
1654
1655
1656
1657 Procedure: GETCURPAT
1658
1659 Called by: M_REGPAT (procedure in DEMO.PRG)
1660
1661
1662
1663 FUNCTION getcurpat
1664 PARAMETERS mpid, mean, mlast, mfirst, mbirth, msex, mrate
1665
1666 * load patient details from global memory
1667 mpid = apatientid
1668 mean = mdocsec
1669 mlast = mlastname
1670 mfirst = mfirstname
1671

```

Page 13 of 37

```

1671 mbirth = mdatebirth
1672 max = msex
1673 mrate = mrate
1674 RETURN mpid
1675
1676 -----
1677 * set current patient info
1678 *
1679 *
1680 -----
1681
1682 Procedure: SETCURPAT
1683
1684 Called by: M_REGPAT (procedure in DEMO.PRG)
1685
1686 *****
1687 FUNCTION setcurpat
1688 PARAMETERS mpid, msn, mlast, mfirst, mbirth, msex, mrate
1689
1690 * save patient details in global memory
1691 mpatientid = mpid
1692 mlastsec = msn
1693 mlastname = mlast
1694 mfirstname = mfirst
1695 mdatebirth = mbirth
1696 msex = msex
1697 mrate = mrate
1698 RETURN mpatientid
1699
1700 * patient screen event procedures
1701
1702 * Select patient from browse list
1703 *****
1704
1705 Procedure: M_PATLIST
1706
1707 Called by: _G700ZD65J() (function in W4.SPR)
1708
1709 Calls: GETPATIENT (procedure in DEMO.PRG)
1710 : M_PATGET (procedure in DEMO.PRG)
1711
1712 *****
1713 PROCEDURE m_patlist
1714 PRIVATE newsn
1715 newsn = getpatient()
1716 IF EMPTY(newsn)
1717 mpatget(mpatientid, mlastsec, mlastname, mfirstname, mdatebirth
1718 : msex, mrate)
1719 RETURN
1720
1721 * GET Existing patient record from database
1722 *****
1723
1724 Procedure: M_PATGET
1725
1726 Called by: ENCOUNT (procedure in DEMO.PRG)
1727 : M_PATLIST (procedure in DEMO.PRG)
1728 : STENCOUNT (procedure in U.PRG)
1729
1730 Calls: PATGETINFO() (function in U.PRG)
1731 : ERRMESSAGE() (function in U.PRG)
1732
1733 *****
1734

```

```

1735 FUNCTION m_patget
1736 PARAMETERS mpid, msn, mlast, mfirst, mbirth, msex, mrate
1737 PRIVATE mevent
1738 mevent = e_ok
1739 IF ipatgetinfo(mpid, msn, mlast, mfirst, mbirth, msex, mrate)
1740 * not found: - this should never happen!
1741 merrmessage("patient record not found [" + msn + "]", m)
1742 mevent = e_esc
1743 ENDIF
1744 RETURN mevent
1745
1746 * PUT updated patient record to database
1747 *****
1748
1749 Procedure: M_PATPUT
1750
1751 Called by: M_REGPAT (procedure in DEMO.PRG)
1752
1753 Calls: POPMESSAGE() (function in U.PRG)
1754 : ISVALIDSSH() (function in U.PRG)
1755 : PATPUTINFO() (function in U.PRG)
1756
1757 *****
1758 FUNCTION m_patput
1759 PARAMETERS mpid, msn, mlast, mfirst, mbirth, msex, mrate
1760 PRIVATE mevent
1761 mevent = e_ok
1762 IF IIF(TYPE("allow_new") = "U", .F., allow_new)
1763 * new patient not authorized
1764 mpopmessage("New patient not allowed", 5)
1765 mevent = e_esc
1766 ELSE
1767 * ensure patient exists
1768 IF IIF(SSH(msn) = "U", .F., allow_new)
1769 mpopmessage("patient not on file [" + msn + "]", m)
1770 mevent = e_esc
1771 ELSE
1772 * update record for patient
1773 mpatputinfo(mpid, msn, mlast, mfirst, mbirth, msex, mrate)
1774 ENDIF
1775 RETURN mevent
1776
1777 * PUT New patient record
1778 *****
1779
1780 Procedure: M_PATADD
1781
1782 Called by: M_REGPAT (procedure in DEMO.PRG)
1783
1784 Calls: POPMESSAGE() (function in U.PRG)
1785 : ISVALIDSSH() (function in U.PRG)
1786 : ADDPAT() (function in U.PRG)
1787 : NEUPATIENT() (function in U.PRG)
1788
1789 *****
1790 FUNCTION m_patadd
1791 PARAMETERS mpid, msn, mlast, mfirst, mbirth, msex, mrate
1792 PRIVATE mevent
1793 mevent = e_ok
1794 IF IIF(TYPE("allow_new") = "U", .F., allow_new)
1795 * new patient not authorized
1796 mpopmessage("New patient not allowed", 5)
1797 mevent = e_esc
1798 ELSE
1799 * ensure patient unique
1800 *****
1801

```

```

1802 IF isvalidsn(msn)
1803   -popmessage("patient already on file [" + msn + "])
1804   mevent = e_esc
1805 ELSE
1806   * create record for new patient      ** add patient id
1807   mpid = addpat()
1808   -newpatient(mpid,msn,m1ast,mfirst,mbirth,max,mrate)
1809   -END IF
1810   -RETURN mevent
1811
1812 *-----
1813 *
1814 * compute patient's age on given date
1815 *
1816 *
1817 *-----
1818 *****
1819 *****
1820 *****
1821 *****
1822 *****
1823 *****
1824 *****
1825 *****
1826 *****
1827 *****
1828 *****
1829 *****
1830 *****
1831 *****
1832 *****
1833 *****
1834 *****
1835 *****
1836 *****
1837 *****
1838 *****
1839 *****
1840 *****
1841 *****
1842 *****
1843 *****
1844 *****
1845 *****
1846 *****
1847 *****
1848 *****
1849 *****
1850 *****
1851 *****
1852 *****
1853 *****
1854 *****
1855 *****
1856 *****
1857 *****
1858 *****
1859 *****
1860 *****
1861 *****
1862 *****
1863 *****
1864 *****
1865 *****
1866 *****
1867 *****

```

```

Procedure: AGEAT
Called by: M_RESUME (procedure in DEMO.PRG)
          M_REGPAT (procedure in DEMO.PRG)
          M_ENGET (procedure in DEMO.PRG)
          M_ENCNEW (procedure in DEMO.PRG)
          WPAT (procedure in DEMO.PRG)
*****
FUNCTION ageat
PARAMETERS mdate, mdatebirth
PRIVATE mege
mege = YEAR(mdate) - YEAR(mdatebirth)
IF MONTH(mdate) < MONTH(mdatebirth)
mege = mege - 1
ELSE
IF MONTH(mdate) = MONTH(mdatebirth) .AND.
DAY(mdate) < DAY(mdatebirth)
mege = mege - 1
ENDIF
ENDIF
IF mege < 16 OR mege > 79
RETURN ""
ENDIF
RETURN LTRIM(STR(mege))
*****

```

```

Procedure: ADDIAGNOS
Called by: GETDIAGNOS() (function in U.PRG)
Calls: DEFFACTOR (procedure in DEMO.PRG)
*****
FUNCTION adddiagnos
PARAMETERS dname, dval
PRIVATE mv, mn
mn = ALLTRIM(dname)
mv = dval
RETURN defactor(mn, mv)
*****

```

```

DEMO.ACT 10-14-92 10:50a

```

```

** ng memory
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930

```

```

* add history data elements to the memory array
*****
Procedure: ADDHISTORY
Called by: EDSEVAGE (procedure in DEMO.PRG)
Calls: DICTONAME() (function in U.PRG)
      ERASEQ (procedure in DEMO.PRG)
      ADDSYMPTOM (procedure in DEMO.PRG)
*****
FUNCTION addhistory
PARAMETERS histname, histval
EXTERNAL ARRAY histname, histval
PRIVATE i, mid, mstr
FOR i = 1 TO ALEN(histname)
mid = dictionary(histname[i])
mstr = histval[i]
IF EMPTY(mid)
-eraseq(mid)
IF EMPTY(mstr)
-addsymptom(mid, mstr)
ENDIF
ENDIF
RETURN .T.
*****

```

```

* add data element to the memory array
*****
Procedure: ADDSYMPTOM
Called by: ADDHISTORY (procedure in DEMO.PRG)
          GETSYMPTOM() (function in U.PRG)
Calls: DEFC (procedure in DEMO.PRG)
*****
FUNCTION addsymptom
PARAMETERS mvname, mval
PRIVATE mv, mn
mn = ALLTRIM(mvname)
mv = ALLTRIM(mval)
-defq(mn, mv)
RETURN mv
*****

```

```

* delete all diagnoses from memory
*****

```

```

1931 Procedure: DELDIAGNOS
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
2002
2003
2004
2005
2006

```

Procedure: DELDIAGNOS

Called by: INITDIAGNOS()
: BATCTEST()
: M ENCHIEV
: L0ENEC

Calls: RESETDIAG
(procedure in DEMO.PRG)

FUNCTION deldiag
mdag = ""
resetdiag()
nextdiag = 1
RETURN ""

Procedure: RESETDIAG

Called by: M ASSIST
: DELDIAGNOS

FUNCTION resetdiag
mconf = 0
RETURN ""

LL reset diagnosis values

* delete all symptoms from memory

Procedure: DELSYMPTOM

Called by: INITSYMPTOM()
: BATCTEST()
: M ENCHIEV
: L0ENEC

FUNCTION delsymptom
mcondition = ""
mymovel = ""
qualstack = ""
redoal = ""
aqual = 1
nextqual = 1
RETURN nextqual

LL erase names
LL erase values
LL redo stack for qualifiers
LL next slot in list of conditions

* Increment confidence factor

Procedure: INCR
Called by: DECR
(procedure in DEMO.PRG)

```

2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000

```

Calls: GETFACTOR
: SETFACTOR
(procedure in DEMO.PRG)
(procedure in DEMO.PRG)

FUNCTION incr
PARAMETERS matr, mincr
PRIVATE M
M = getfactor(matr)
RETURN setfactor(matr, M + mincr)

* Decrement confidence factor

Procedure: DECR
Called by: INCR
(procedure in DEMO.PRG)

FUNCTION decr
PARAMETERS matr, mincr
RETURN incr(matr, -mincr)

* Multiply confidence factor

Procedure: MULFACTOR
Called by: EVALCOMO
(procedure in DEMO.PRG)
Called by: FACTORID
(procedure in DEMO.PRG)

FUNCTION mulfactor
PARAMETERS matr, mval
PRIVATE M
PRIVATE I
I = factorid(matr) LL search for name of factor
M = mconf(I)
M = IIF(M = 0, M * mval, mval)
mconf(I) = M
RETURN M

LL rdd07
LL rdd07
LL rdd07
LL rdd07
LL rdd07

* Sum confidence factors

Procedure: SUMFACTOR
Called by: MODFACTOR
(procedure in DEMO.PRG)

FUNCTION sumfactor

```

2063 PRIVATE M
2064 PRIVATE I
2065 M = 0
2066
2067 FOR I = 1 TO nextdiag - 1
2068   M = M + mconf(i)
2069   LNEXT
2070
2071 RETURN M
2072
2073
2074
2075
2076 * Normalize confidence factors
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092 Procedure: NORMFACTOR
2093
2094 Called by: MAKEDIAG (procedure in DEMO.PRG)
2095
2096 Calls: SUMFACTOR (procedure in DEMO.PRG)
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137

```

```

2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193

```

```

2194 i = nextdiag
2195 =AINS(mdiag, i)
2196 =AINS(mconf, i)
2197 mdiag(i) = matr
2198 mconf(i) = 0
2199 nextdiag = nextdiag + 1
2200 ENDIF
2201 IF nextdiag = "OFF"
2202 SET EXACT OFF
2203 ENDIF
2204 RETURN i
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309

```

i = nextdiag
 =AINS(mdiag, i)
 =AINS(mconf, i)
 mdiag(i) = matr
 mconf(i) = 0
 nextdiag = nextdiag + 1
 ENDIF
 IF nextdiag = "OFF"
 SET EXACT OFF
 ENDIF
 RETURN i

Procedure: ADDFACTOR
 Calls: FACTORID (procedure in DEMO.PRG)

FUNCTION addfactor
 PARAMETERS matr
 PRIVATE i, savef
 i = factorid(matr) && get identifier
 savef = matr + arrow + IIF(mconf(i) = 0, uval, STR(mco
 => nft(i))
 IF TYPE("statevecf[arule]") = "M"
 statevecf[arule] = ""
 ENDIF
 statevecf[arule] = statevecf[arule] + savef + eol
 RETURN .T.

Procedure: INITDIAGNOSIS
 Calls: DISEASECNT (function in U.PRG)
 : DELDIAGNOS (procedure in DEMO.PRG)

FUNCTION initdiag
 PUBLIC ndiseases
 ndiseases = diseasecnt()
 && maximum pos
 && number of n

Procedure: LOOKUPQ
 Calls: DICTO (function in U.PRG)

FUNCTION lookupq
 PARAMETERS mid
 PRIVATE mq, mtype, mstable
 mq = look up identifier in dictionary
 mtype = return associated question text
 mstable = .f.
 && dictq[mid, mstable, mq]
 IF EMPTY(mtype)

```

2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309

```

RETURN ndiseases
 * initialize array of symptoms
 FUNCTION: INITSYMPTOM()
 Called by: DEMO.PRG
 Calls: DICTCOUNT (function in U.PRG)
 : DELSYMPTOM (procedure in DEMO.PRG)

FUNCTION initSYMPTOM
 PUBLIC nconditions
 nconditions = dictcount()
 && maximum poss
 && number of num

* create array of symptoms/conditions for current encounter
 PUBLIC mcondition[nconditions]
 PUBLIC msymptval[nconditions]
 PUBLIC qualstack[nconditions]
 PUBLIC qualval[nconditions]
 PUBLIC redoval[nconditions]
 PUBLIC mqual
 PUBLIC nextqual
 =delSYMPTOM()
 RETURN nconditions

* Lookup Qualifier
 Procedure: LOOKUPQ
 Called by: CHANGED
 : LASTQ
 Calls: DICTO (function in U.PRG)

FUNCTION lookupq
 PARAMETERS mid
 PRIVATE mq, mtype, mstable
 mq = look up identifier in dictionary
 mtype = return associated question text
 mstable = .f.
 && dictq[mid, mstable, mq]
 IF EMPTY(mtype)


```

2310 IF mtype = "X" .OR. !maskable
2311 * prepend meta symbol
2312 mq = "M" + mq
2313 ENDIF
2314 RETURN mq
2315
2316 *****
2317
2318 Procedure: LOOKUPM
2319
2320 Called by: STRCOND (procedure in DEMO.PRG)
2321
2322 Calls: DICTQ (function in U.PRG)
2323
2324 *****
2325
2326 FUNCTION lookuqm
2327
2328 PARAMETERS mid
2329 PRIVATE mq, mtype, maskable, mname
2330 * look up identifier in dictionary
2331 * return associated name
2332 mq = ""
2333 mname = ""
2334 maskable = .F.
2335 mtype = dictq(mid, maskable, mq, mname)
2336 IF !EMPTY(mtype)
2337 * IF mtype = "X" .OR. !maskable
2338 * prepend meta symbol
2339 mname = "M" + mname
2340 ENDIF
2341 RETURN mname
2342
2343 *****
2344
2345 * ask user for value of qualifier
2346
2347 *****
2348
2349 Procedure: ASK
2350
2351 Called by: M.REDOQ (procedure in DEMO.PRG)
2352 : EVALCOND (procedure in DEMO.PRG)
2353
2354 Calls: EVALQ (procedure in DEMO.PRG)
2355
2356 *****
2357
2358 FUNCTION ask
2359
2360 PARAMETERS mstr, mq
2361 * mstr is name of string qualifier
2362 * mq is previous/new value of qualifier
2363 * forces the question to be asked and the qualifier to be
2364 * instantiated.
2365 PRIVATE mresult
2366 IF event = e_esc
2367 * Note! the flag, event is set to e_esc by user hittin
2368
2369 * the abort key during data entry, but a rule may be i
2370 * middle of a complex expression evaluation. Therefore
2371 * code prevents any more questions until the end of th
2372 * rule.
2373
2374 *****

```

```

2372 *****
2373
2374 ELSE
2375 * ensure value is actually valid:
2376 mresult = !IF(event = e_esc, .F., .T.)
2377 ENDIF
2378 RETURN mresult
2379
2380 *****
2381
2382 * test qualifier mstr for occurrence of mval
2383
2384 *****
2385
2386 Procedure: ISO
2387
2388 Calls: EVALQ (procedure in DEMO.PRG)
2389
2390 *****
2391
2392 FUNCTION isoq
2393
2394 PARAMETERS mstr, mval
2395 * mstr is name of string qualifier
2396 * mval is string value to test
2397 PRIVATE mq, mresult
2398 PRIVATE N, mitem
2399 IF event = e_esc
2400 * Note! the flag, event is set to e_esc by user hittin
2401
2402 * the abort key during data entry, but a rule may be i
2403 * middle of a complex expression evaluation. Therefore
2404 * code prevents any more questions until the end of th
2405 * rule.
2406 mresult = .F.
2407 ELSE
2408 mq = evalq(mstr)
2409
2410 *****
2411
2412 IF mresult = mval $ mq
2413
2414 *****
2415
2416 IF mresult
2417 * partial match found, look for exact item
2418 mresult = .F.
2419 * extract names from qualifier
2420 DO WHILE LEN(mq) > 0
2421 N = AT(mconcat, mq) - 1
2422
2423 *****
2424
2425 IF N > 0
2426 mitem = LEFT(mq, N)
2427 ELSE
2428 mitem = mq
2429 ENDIF
2430 IF mitem = mval
2431 mresult = .T.
2432 ENDIF
2433 IF N <= 0
2434 EXIT
2435
2436 *****
2437
2438 IF mresult = SUBSTR(mq, N+2)
2439
2440 *****

```



```

2554 mwidth = 0
2555 mdec = 0
2556 munits = ""
2557 IF dictq(dictid, amhi, amlo, amwidth, amdec, am
    => units)
2558 THEN
2559   dth for data entry
2560   mwidth = MAX(mwidth, 9)
2561   mdec = MAX(mdec, 9)
2562   munits = MAX(munits, 9)
2563   mwidth = MAX(mwidth, 9)
2564   mdec = MAX(mdec, 9)
2565   munits = MAX(munits, 9)
2566   mwidth = MAX(mwidth, 9)
2567   mdec = MAX(mdec, 9)
2568   munits = MAX(munits, 9)
2569   mwidth = MAX(mwidth, 9)
2570   mdec = MAX(mdec, 9)
2571   munits = MAX(munits, 9)
2572   mwidth = MAX(mwidth, 9)
2573   mdec = MAX(mdec, 9)
2574   munits = MAX(munits, 9)
2575   mwidth = MAX(mwidth, 9)
2576   mdec = MAX(mdec, 9)
2577   munits = MAX(munits, 9)
2578   mwidth = MAX(mwidth, 9)
2579   mdec = MAX(mdec, 9)
2580   munits = MAX(munits, 9)
2581   mwidth = MAX(mwidth, 9)
2582   mdec = MAX(mdec, 9)
2583   munits = MAX(munits, 9)
2584   mwidth = MAX(mwidth, 9)
2585   mdec = MAX(mdec, 9)
2586   munits = MAX(munits, 9)
2587   mwidth = MAX(mwidth, 9)
2588   mdec = MAX(mdec, 9)
2589   munits = MAX(munits, 9)
2590   mwidth = MAX(mwidth, 9)
2591   mdec = MAX(mdec, 9)
2592   munits = MAX(munits, 9)
2593   mwidth = MAX(mwidth, 9)
2594   mdec = MAX(mdec, 9)
2595   munits = MAX(munits, 9)
2596   mwidth = MAX(mwidth, 9)
2597   mdec = MAX(mdec, 9)
2598   munits = MAX(munits, 9)
2599   mwidth = MAX(mwidth, 9)
2600   mdec = MAX(mdec, 9)
2601   munits = MAX(munits, 9)
2602   mwidth = MAX(mwidth, 9)
2603   mdec = MAX(mdec, 9)
2604   munits = MAX(munits, 9)
2605   mwidth = MAX(mwidth, 9)
2606   mdec = MAX(mdec, 9)
2607   munits = MAX(munits, 9)

```

```

2607   mwidth = MAX(mwidth, 9)
2608   mdec = MAX(mdec, 9)
2609   munits = MAX(munits, 9)
2610   mwidth = MAX(mwidth, 9)
2611   mdec = MAX(mdec, 9)
2612   munits = MAX(munits, 9)
2613   mwidth = MAX(mwidth, 9)
2614   mdec = MAX(mdec, 9)
2615   munits = MAX(munits, 9)
2616   mwidth = MAX(mwidth, 9)
2617   mdec = MAX(mdec, 9)
2618   munits = MAX(munits, 9)
2619   mwidth = MAX(mwidth, 9)
2620   mdec = MAX(mdec, 9)
2621   munits = MAX(munits, 9)
2622   mwidth = MAX(mwidth, 9)
2623   mdec = MAX(mdec, 9)
2624   munits = MAX(munits, 9)
2625   mwidth = MAX(mwidth, 9)
2626   mdec = MAX(mdec, 9)
2627   munits = MAX(munits, 9)
2628   mwidth = MAX(mwidth, 9)
2629   mdec = MAX(mdec, 9)
2630   munits = MAX(munits, 9)
2631   mwidth = MAX(mwidth, 9)
2632   mdec = MAX(mdec, 9)
2633   munits = MAX(munits, 9)
2634   mwidth = MAX(mwidth, 9)
2635   mdec = MAX(mdec, 9)
2636   munits = MAX(munits, 9)
2637   mwidth = MAX(mwidth, 9)
2638   mdec = MAX(mdec, 9)
2639   munits = MAX(munits, 9)
2640   mwidth = MAX(mwidth, 9)
2641   mdec = MAX(mdec, 9)
2642   munits = MAX(munits, 9)
2643   mwidth = MAX(mwidth, 9)
2644   mdec = MAX(mdec, 9)
2645   munits = MAX(munits, 9)
2646   mwidth = MAX(mwidth, 9)
2647   mdec = MAX(mdec, 9)
2648   munits = MAX(munits, 9)
2649   mwidth = MAX(mwidth, 9)
2650   mdec = MAX(mdec, 9)
2651   munits = MAX(munits, 9)
2652   mwidth = MAX(mwidth, 9)
2653   mdec = MAX(mdec, 9)
2654   munits = MAX(munits, 9)
2655   mwidth = MAX(mwidth, 9)
2656   mdec = MAX(mdec, 9)
2657   munits = MAX(munits, 9)

```

```

2658 PARAMETERS matr, mval
2659 PRIVATE newval
2660 IF event <> e_esc
2661 * Motel the flag, event is set to e_esc by user
2662 => hitting
2663 * the abort key during data entry, but a rule me
2664 * middle of a complex expression evaluation. The
2665 * code prevents any instantiations until the end
2666 * rule.
2667 i = qid(matr)  && get identifier for qualifier
2668 IF EMPTY(msynpval[i])
2669 * qualifier undefined, define new value
2670 msynpval[i] = mval
2671 ELSE
2672 IF LEN(mval) > 0 .AND. .NOT. mval $ msynpval[
2673 => i]
2674 * qualifier does not already contain this
2675 msynpval[i] = msynpval[i] + mconcat + mval
2676 ENDIF
2677 newval = msynpval[i]
2678 redoal[i] = newval  && save on redo
2679 ELSE
2680 * suppress instantiation, return null value
2681 newval = ""
2682 ENDIF
2683 RETURN newval
2684
2685
2686 * Get Qualifier
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000

```

```

2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900

```



```

2948 L-NDIF statevecq[arule] = statevecq[arule] + saveq +
2949 =>
2950 eol
2951 L-NDIF
2952 RETURN .T.
2953
2954
2955 *
2956 * get conditions to string form
2957 *
2958
2959 *****
2960 Procedure: STRCOND
2961
2962 Called by: M RESUME
2963           : DEMO.PRG
2964           : U.PRG
2965           : EDITNAMP
2966
2967 Calls: LOOKUPNAM
2968           : DEMO.PRG
2969 *****
2970
2971 FUNCTION strcond
2972
2973 PARAMETERS second, mqid, arname, mval, DELIM
2974
2975 EXTERNAL ARRAY arname
2976
2977
2978
2979
2980
2981 PRIVATE i, j, N, DELIM
2982 PRIVATE mstr
2983
2984 IF PARAMETERS() = 4
2985   DELIM = " "
2986   ENDIF
2987   N = nextqual - 1
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
373
```

```

3016 *|
3017 *| Procedure: SYNDIAG
3018 *|
3019 *| Called by: M.RESUME (procedure in DEMO.PRG)
3020 *| : _EDITASSIS (procedure in U.PRG)
3021 *|
3022 *| Calls: DISEASENAM() (function in U.PRG)
3023 *|
3024 *****
3025 FUNCTION strdiag
3026 *****
3027 PARAMETERS msorted, mdlag, mconf, threshold,
3028 *****
3029 EXTERNAL ARRAY msorted
3030 *****
3031 PRIVATE i, DELIM, K, inserted, msorted, xnum
3032 *****
3033 DIMENSION mrank( nextdiag )
3034 *****
3035 inserted = 0
3036 *****
3037 DELIM = " "
3038 *****
3039 N = nextdiag - 1
3040 *****
3041 FOR i = 1 TO N
3042 *****
3043   = AINS(mrank, i)
3044 *****
3045   IF mconf[i] >= threshold .AND.
3046     IF( TYPE("limit") < "M", .T., .F., mconf
3047     * factor is within specified range
3048     * insert in ranked list
3049     K = 1
3050 *****
3051   DO WHILE K <= inserted
3052     * find slot in ranked list of factor
3053 *****
3054     IF mconf[i] >= mconf(mrank(k))
3055       * insert factor
3056       = AINS(mrank, K)
3057       EXIT
3058     * K = K + 1
3059 *****
3060     mrank(k) = 1
3061     inserted = inserted + 1
3062 *****
3063   ENDIF
3064   NEXT
3065 *****
3066   * output ranked list as string
3067 *****
3068   FOR i = 1 TO inserted
3069     K = mrank(i)
3070     msorted(i) = diseaseNAM( VAL( mdlag(k)
3071 *****
3072   IF( PARAMETERS() = 6, ;
3073     DELIM = LTRIM( STR( mconf( k ), 6
3074 *****
3075   )
3076 *****
3077   )
3078 *****
3079   RETURN inserted
3080 *****

```

```

3061 *****
3062 *****
3063 *****
3064 *****
3065 *****
3066 *****
3067 *****
3068 *****
3069 *****
3070 *****
3071 *****
3072 *****
3073 *****
3074 *****
3075 *****
3076 *****
3077 *****
3078 *****
3079 *****
3080 *****
3081 *****
3082 *****
3083 *****
3084 *****
3085 *****
3086 *****
3087 *****
3088 *****
3089 *****
3090 *****
3091 *****
3092 *****
3093 *****
3094 *****
3095 *****
3096 *****
3097 *****
3098 *****
3099 *****
3100 *****
3101 *****
3102 *****
3103 *****
3104 *****
3105 *****
3106 *****
3107 *****
3108 *****
3109 *****
3110 *****
3111 *****
3112 *****
3113 *****
3114 *****
3115 *****
3116 *****
3117 *****
3118 *****
3119 *****
3120 *****

```

Procedure: STRNATURAL
 Calls: GETO
 : DICTREPORT()
 : SPLIT()
 Uses: PHRASE.DBF
 : SENTENCE.DBF
 : SECTION.DBF
 : REPORT.DBF
 Indexes: PHRASE.IDX
 : SENTENCE.IDX
 : SECTION.IDX
 : REPORT.IDX

FUNCTION STRNATURAL
 PARAMETERS marea
 PRIVATE mstr, xstr, ystr, zstr
 PRIVATE mid
 PRIVATE msection, mstatement, mabing
 PRIVATE msel
 msel = SELECT()
 ** save curr id database selection
 SELECT 0
 USE PHRASE INDEX PHRASE
 SELECT 0
 USE SENTENCE INDEX SENTENCE
 SET RELATION TO id INTO dict, id INTO ph
 SELECT 0
 USE SECTION INDEX SECTION
 SET RELATION TO sentence INTO sentence
 SELECT 0
 USE REPORT INDEX REPORT
 SET RELATION TO SECTION INTO SECTION
 SET FILTER TO area = marea
 GOTO TOP
 mstr = ""
 DO WHILE !EOF()
 msection = SECTION
 SELECT SECTION
 mabing = .1.
 ystr = ""
 DO WHILE SECTION = report->section
 * test controlling condition
 DO CASE
 CASE OBJECT = "S"
 mstr = GETQC ALLTRIM(STR(id


```

3328      77 "
3329      j = j + 1
3330      ELSE
3331      ? ALLTRIM( STR( mercoutid ) )
3332      77 " was incomplete ***
3333      *** Expected NEXT--found ENDOF
3334      L-ENDDIF
3335      IF hasdata
3336      REPLACE hasdata WITH .T., case
3337      L-ENDDIF
3338      REPLACE outcome WITH szoutcome
3339      K = K + 1
3340      SKIP
3341      L-ENDDO
3342      *** Expected ENDOF--found ENDDO
3343      ? "Total cases read: " + ALLTRIM( S
3344      ? "Total completed: " + ALLTRIM( S
3345      L-ENDDO
3346      IF EMPTY(logfile)
3347      SET PRINTER OFF
3348      SET PRINTER TO
3349      SET CONSOLE ON
3350      L-ENDDIF
3351      ? "Hit any key to continue..."
3352      =INKEY(0)
3353      RETURN .T.
3354      *
3355      *
3356      *
3357      *
3358      *
3359      *
3360      *
3361      *
3362      *
3363      *
3364      *
3365      *
3366      *
3367      *
3368      *
3369      *
3370      *
3371      *
3372      *
3373      *
3374      *
3375      *
3376      *
3377      *
3378      *
3379      *
3380      *
3381      *
3382      *
3383      *
3384      *
3385      *
3386      *
3387      *
3388      *
3389      *
3390      *
3391      *
3392      *
3393      *
3394      *
3395      *
3396      *
3397      *
3398      *
3399      *
3400      *
3401      *
3402      *
3403      *
3404      *
3405      *
3406      *
3407      *
3408      *
3409      *
3410      *
3411      *
3412      *
3413      *
3414      *
3415      *
3416      *
3417      *
3418      *
3419      *
3420      *
3421      *
3422      *
3423      *
3424      *
3425      *
3426      *
3427      *
3428      *
3429      *
3430      *
3431      *
3432      *
3433      *
3434      *
3435      *
3436      *
3437      *
3438      *
3439      *
3440      *
3441      *
3442      *
3443      *
3444      *
3445      *
3446      *
3447      *
3448      *
3449      *
3450      *
3451      *
3452      *
3453      *
3454      *
3455      *
3456      *
3457      *
3458      *
3459      *
3460      *
3461      *
3462      *
3463      *
3464      *
3465      *
3466      *
3467      *
3468      *
3469      *
3470      *
3471      *
3472      *
3473      *
3474      *
3475      *
3476      *
3477      *
3478      *
3479      *
3480      *
3481      *
3482      *
3483      *
3484      *
3485      *
3486      *
3487      *
3488      *
3489      *
3490      *
3491      *
3492      *
3493      *
3494      *
3495      *
3496      *
3497      *
3498      *
3499      *
3500      *
3501      *
3502      *
3503      *
3504      *
3505      *
3506      *
3507      *
3508      *
3509      *
3510      *
3511      *
3512      *
3513      *
3514      *
3515      *
3516      *
3517      *
3518      *
3519      *
3520      *
3521      *
3522      *
3523      *
3524      *
3525      *
3526      *
3527      *
3528      *
3529      *
3530      *
3531      *
3532      *
3533      *
3534      *
3535      *
3536      *
3537      *
3538      *
3539      *
3540      *
3541      *
3542      *
3543      *
3544      *
3545      *
3546      *
3547      *
3548      *
3549      *
3550      *
3551      *
3552      *
3553      *
3554      *
3555      *
3556      *
3557      *
3558      *
3559      *
3560      *
3561      *
3562      *
3563      *
3564      *
3565      *
3566      *
3567      *
3568      *
3569      *
3570      *
3571      *
3572      *
3573      *
3574      *
3575      *
3576      *
3577      *
3578      *
3579      *
3580      *
3581      *
3582      *
3583      *
3584      *
3585      *
3586      *
3587      *
3588      *
3589      *
3590      *
3591      *
3592      *
3593      *
3594      *
3595      *
3596      *
3597      *
3598      *
3599      *
3600      *
3601      *
3602      *
3603      *
3604      *
3605      *
3606      *
3607      *
3608      *
3609      *
3610      *
3611      *
3612      *
3613      *
3614      *
3615      *
3616      *
3617      *
3618      *
3619      *
3620      *
3621      *
3622      *
3623      *
3624      *
3625      *
3626      *
3627      *
3628      *
3629      *
3630      *
3631      *
3632      *
3633      *
3634      *
3635      *
3636      *
3637      *
3638      *
3639      *
3640      *
3641      *
3642      *
3643      *
3644      *
3645      *
3646      *
3647      *
3648      *
3649      *
3650      *
3651      *
3652      *
3653      *
3654      *
3655      *
3656      *
3657      *
3658      *
3659      *
3660      *
3661      *
3662      *
3663      *
3664      *
3665      *
3666      *
3667      *
3668      *
3669      *
3670      *
3671      *
3672      *
3673      *
3674      *
3675      *
3676      *
3677      *
3678      *
3679      *
3680      *
3681      *
3682      *
3683      *
3684      *
3685      *
3686      *
3687      *
3688      *
3689      *
3690      *
3691      *
3692      *
3693      *
3694      *
3695      *
3696      *
3697      *
3698      *
3699      *
3700      *
3701      *
3702      *
3703      *
3704      *
3705      *
3706      *
3707      *
3708      *
3709      *
3710      *
3711      *
3712      *
3713      *
3714      *
3715      *
3716      *
3717      *
3718      *
3719      *
3720      *
3721      *
3722      *
3723      *
3724      *
3725      *
3726      *
3727      *
3728      *
3729      *
3730      *
3731      *
3732      *
3733      *
3734      *
3735      *
3736      *
3737      *
3738      *
3739      *
3740      *
3741      *
3742      *
3743      *
3744      *
3745      *
3746      *
3747      *
3748      *
3749      *
3750      *
3751      *
3752      *
3753      *
3754      *
3755      *
3756      *
3757      *
3758      *
3759      *
3760      *
3761      *
3762      *
3763      *
3764      *
3765      *
3766      *
3767      *
3768      *
3769      *
3770      *
3771      *
3772      *
3773      *
3774      *
3775      *
3776      *
3777      *
3778      *
3779      *
3780      *
3781      *
3782      *
3783      *
3784      *
3785      *
3786      *
3787      *
3788      *
3789      *
3790      *
3791      *
3792      *
3793      *
3794      *
3795      *
3796      *
3797      *
3798      *
3799      *
3800      *
3801      *
3802      *
3803      *
3804      *
3805      *
3806      *
3807      *
3808      *
3809      *
3810      *
3811      *
3812      *
3813      *
3814      *
3815      *
3816      *
3817      *
3818      *
3819      *
3820      *
3821      *
3822      *
3823      *
3824      *
3825      *
3826      *
3827      *
3828      *
3829      *
3830      *
3831      *
3832      *
3833      *
3834      *
3835      *
3836      *
3837      *
3838      *
3839      *
3840      *
3841      *
3842      *
3843      *
3844      *
3845      *
3846      *
3847      *
3848      *
3849      *
3850      *
3851      *
3852      *
3853      *
3854      *
3855      *
3856      *
3857      *
3858      *
3859      *
3860      *
3861      *
3862      *
3863      *
3864      *
3865      *
3866      *
3867      *
3868      *
3869      *
3870      *
3871      *
3872      *
3873      *
3874      *
3875      *
3876      *
3877      *
3878      *
3879      *
3880      *
3881      *
3882      *
3883      *
3884      *
3885      *
3886      *
3887      *
3888      *
3889      *
3890      *
3891      *
3892      *
3893      *
3894      *
3895      *
3896      *
3897      *
3898      *
3899      *
3900      *
3901      *
3902      *
3903      *
3904      *
3905      *
3906      *
3907      *
3908      *
3909      *
3910      *
3911      *
3912      *
3913      *
3914      *
3915      *
3916      *
3917      *
3918      *
3919      *
3920      *
3921      *
3922      *
3923      *
3924      *
3925      *
3926      *
3927      *
3928      *
3929      *
3930      *
3931      *
3932      *
3933      *
3934      *
3935      *
3936      *
3937      *
3938      *
3939      *
3940      *
3941      *
3942      *
3943      *
3944      *
3945      *
3946      *
3947      *
3948      *
3949      *
3950      *
3951      *
3952      *
3953      *
3954      *
3955      *
3956      *
3957      *
3958      *
3959      *
3960      *
3961      *
3962      *
3963      *
3964      *
3965      *
3966      *
3967      *
3968      *
3969      *
3970      *
3971      *
3972      *
3973      *
3974      *
3975      *
3976      *
3977      *
3978      *
3979      *
3980      *
3981      *
3982      *
3983      *
3984      *
3985      *
3986      *
3987      *
3988      *
3989      *
3990      *
3991      *
3992      *
3993      *
3994      *
3995      *
3996      *
3997      *
3998      *
3999      *
4000      *

```

```

4001      *
4002      *
4003      *
4004      *
4005      *
4006      *
4007      *
4008      *
4009      *
4010      *
4011      *
4012      *
4013      *
4014      *
4015      *
4016      *
4017      *
4018      *
4019      *
4020      *
4021      *
4022      *
4023      *
4024      *
4025      *
4026      *
4027      *
4028      *
4029      *
4030      *
4031      *
4032      *
4033      *
4034      *
4035      *
4036      *
4037      *
4038      *
4039      *
4040      *
4041      *
4042      *
4043      *
4044      *
4045      *
4046      *
4047      *
4048      *
4049      *
4050      *
4051      *
4052      *
4053      *
4054      *
4055      *
4056      *
4057      *
4058      *
4059      *
4060      *
4061      *
4062      *
4063      *
4064      *
4065      *
4066      *
4067      *
4068      *
4069      *
4070      *
4071      *
4072      *
4073      *
4074      *
4075      *
4076      *
4077      *
4078      *
4079      *
4080      *
4081      *
4082      *
4083      *
4084      *
4085      *
4086      *
4087      *
4088      *
4089      *
4090      *
4091      *
4092      *
4093      *
4094      *
4095      *
4096      *
4097      *
4098      *
4099      *
4100      *
4101      *
4102      *
4103      *
4104      *
4105      *
4106      *
4107      *
4108      *
4109      *
4110      *
4111      *
4112      *
4113      *
4114      *
4115      *
4116      *
4117      *
4118      *
4119      *
4120      *
4121      *
4122      *
4123      *
4124      *
4125      *
4126      *
4127      *
4128      *
4129      *
4130      *
4131      *
4132      *
4133      *
4134      *
4135      *
4136      *
4137      *
4138      *
4139      *
4140      *
4141      *
4142      *
4143      *
4144      *
4145      *
4146      *
4147      *
4148      *
4149      *
4150      *
4151      *
4152      *
4153      *
4154      *
4155      *
4156      *
4157      *
4158      *
4159      *
4160      *
4161      *
4162      *
4163      *
4164      *
4165      *
4166      *
4167      *
4168      *
4169      *
4170      *
4171      *
4172      *
4173      *
4174      *
4175      *
4176      *
4177      *
4178      *
4179      *
4180      *
4181      *
4182      *
4183      *
4184      *
4185      *
4186      *
4187      *
4188      *
4189      *
4190      *
4191      *
4192      *
4193      *
4194      *
4195      *
4196      *
4197      *
4198      *
4199      *
4200      *
4201      *
4202      *
4203      *
4204      *
4205      *
4206      *
4207      *
4208      *
4209      *
4210      *
4211      *
4212      *
4213      *
4214      *
4215      *
4216      *
4217      *
4218      *
4219      *
4220      *
4221      *
4222      *
4223      *
4224      *
4225      *
4226      *
4227      *
4228      *
4229      *
4230      *
4231      *
4232      *
4233      *
4234      *
4235      *
4236      *
4237      *
4238      *
4239      *
4240      *
4241      *
4242      *
4243      *
4244      *
4245      *
4246      *
4247      *
4248      *
4249      *
4250      *
4251      *
4252      *
4253      *
4254      *
4255      *
4256      *
4257      *
4258      *
4259      *
4260      *
4261      *
4262      *
4263      *
4264      *
4265      *
4266      *
4267      *
4268      *
4269      *
4270      *
4271      *
4272      *
4273      *
4274      *
4275      *
4276      *
4277      *
4278      *
4279      *
4280      *
4281      *
4282      *
4283      *
4284      *
4285      *
4286      *
4287      *
4288      *
4289      *
4290      *
4291      *
4292      *
4293      *
4294      *
4295      *
4296      *
4297      *
4298      *
4299      *
4300      *
4301      *
4302      *
4303      *
4304      *
4305      *
4306      *
4307      *
4308      *
4309      *
4310      *
4311      *
4312      *
4313      *
4314      *
4315      *
4316      *
4317      *
4318      *
4319      *
4320      *
4321      *
4322      *
4323      *
4324      *
4325      *
4326      *
4327      *
4328      *
4329      *
4330      *
4331      *
4332      *
4333      *
4334      *
4335      *
4336      *
4337      *
4338      *
4339      *
4340      *
4341      *
4342      *
4343      *
4344      *
4345      *
4346      *
4347      *
4348      *
4349      *
4350      *
4351      *
4352      *
4353      *
4354      *
4355      *
4356      *
4357      *
4358      *
4359      *
4360      *
4361      *
4362      *
4363      *
4364      *
4365      *
4366      *
4367      *
4368      *
4369      *
4370      *
4371      *
4372      *
4373      *
4374      *
4375      *
4376      *
4377      *
4378      *
4379      *
4380      *
4381      *
4382      *
4383      *
4384      *
4385      *
4386      *
4387      *
4388      *
4389      *
4390      *
4391      *
4392      *
4393      *
4394      *
4395      *
4396      *
4397      *
4398      *
4399      *
4400      *
4401      *
4402      *
4403      *
4404      *
4405      *
4406      *
4407      *
4408      *
4409      *
4410      *
4411      *
4412      *
4413      *
4414      *
4415      *
4416      *
4417      *
4418      *
4419      *
4420      *
4421      *
4422      *
4423      *
4424      *
4425      *
4426      *
4427      *
4428      *
4429      *
4430      *
4431      *
4432      *
4433      *
4434      *
4435      *
4436      *
4437      *
4438      *
4439      *
4440      *
4441      *
4442      *
4443      *
4444      *
4445      *
4446      *
4447      *
4448      *
4449      *
4450      *
4451      *
4452      *
4453      *
4454      *
4455      *
4456      *
4457      *
4458      *
4459      *
4460      *
4461      *
4462      *
4463      *
4464      *
4465      *
4466      *
4467      *
4468      *
4469      *
4470      *
4471      *
4472      *
4473      *
4474      *
4475      *
4476      *
4477      *
4478      *
4479      *
4480      *
4481      *
4482      *
4483      *
4484      *
4485      *
4486      *
4487      *
4488      *
4489      *
4490      *
4491      *
4492      *
4493      *
4494      *
4495      *
4496      *
4497      *
4498      *
4499      *
4500      *

```

DEMO. ACT	10-14-92	10:50a

Page 29 of 37


```

3761 * : SF5YMP (procedure in DEMO.PRG)
3762 * : POPUPHIDE() (function in U.PRG)
3763 * : SFDIAG (procedure in DEMO.PRG)
3764 * : SFTREAT (procedure in DEMO.PRG)
3765 * : EDITTEXT (procedure in DEMO.PRG)
3766 *
3767 *****
3753 FUNCTION sfreport
3754 PARAMETER marea
3755 mtext = ""
3756 mgenerate = .f.
3757 IF EMPTY(mreport)
3758 mgenerate = yesno("SF600
3759 "E-GENERATE")
3760 ENDIF
3761 IF mgenerate
3762 winpop = popupshow("Generat
3763 newline = CHR(13) + CHR(10)
3764 mearea = marea
3765 mtext = sf5ymp(mearea)
3766 maccess = sfdiag()
3767 mtreas = sftreat()
3768 IF EMPTY(mtext)
3769 mtext = IF(EMPTY(macce
3770 ss), mtext+maccess, mtext)
3771 t, mtext+mtreas, mtext)
3772 ELSE
3773 mtext = mreport
3774 ENDIF
3775 IF EMPTY(mtext)
3776 edittext(mtext)
3777 ENDIF
3778 mreport = mtext
3779 RETURN
3780 *****
3781 * Symptom Report session in na
3782 * Programmer: HLL
3783 *****
3784 *****
3785 *****
3786 *****
3787 *****
3788 *****
3789 *****
3790 *****
3791 *****
3792 *****
3793 *****
3794 *****
3795 *****
3796 *****
3797 *****
3798 *****
3799 *****
3800 *****
3801 *****
3802 *****
3803 *****
3804 *****
3805 *****
3806 *****
3807 *****
3808 *****
3809 *****
3810 *****
3811 *****
3812 *****
3813 *****
3814 *****
3815 *****
3816 *****
3817 *****
3818 *****
3819 *****
3820 *****
3821 *****
3822 *****
3823 *****
3824 *****
3825 *****
3826 *****
3827 *****
3828 *****
3829 *****
3830 *****
3831 *****
3832 *****
3833 *****
3834 *****
3835 *****
3836 *****
3837 *****
3838 *****
3839 *****
3840 *****
3841 *****
3842 *****
3843 *****
3844 *****
3845 *****
3846 *****
3847 *****
3848 *****
3849 *****
3850 *****
3851 *****
3852 *****
3853 *****
3854 *****
3855 *****
3856 *****
3857 *****
3858 *****
3859 *****
3860 *****
3861 *****
3862 *****
3863 *****
3864 *****
3865 *****
3866 *****
3867 *****
3868 *****
3869 *****
3870 *****
3871 *****
3872 *****
3873 *****
3874 *****
3875 *****
3876 *****
3877 *****
3878 *****
3879 *****
3880 *****
3881 *****
3882 *****
3883 *****
3884 *****
3885 *****
3886 *****
3887 *****
3888 *****
3889 *****
3890 *****
3891 *****
3892 *****
3893 *****
3894 *****
3895 *****
3896 *****
3897 *****
3898 *****
3899 *****
3900 *****

```

```

3805 *****
3806 *****
3807 *****
3808 *****
3809 *****
3810 *****
3811 *****
3812 *****
3813 *****
3814 *****
3815 *****
3816 *****
3817 *****
3818 *****
3819 *****
3820 *****
3821 *****
3822 *****
3823 *****
3824 *****
3825 *****
3826 *****
3827 *****
3828 *****
3829 *****
3830 *****
3831 *****
3832 *****
3833 *****
3834 *****
3835 *****
3836 *****
3837 *****
3838 *****
3839 *****
3840 *****
3841 *****
3842 *****
3843 *****
3844 *****
3845 *****
3846 *****
3847 *****
3848 *****
3849 *****
3850 *****
3851 *****
3852 *****
3853 *****
3854 *****
3855 *****
3856 *****
3857 *****
3858 *****
3859 *****
3860 *****
3861 *****
3862 *****
3863 *****
3864 *****
3865 *****
3866 *****
3867 *****
3868 *****
3869 *****
3870 *****
3871 *****
3872 *****
3873 *****
3874 *****
3875 *****
3876 *****
3877 *****
3878 *****
3879 *****
3880 *****
3881 *****
3882 *****
3883 *****
3884 *****
3885 *****
3886 *****
3887 *****
3888 *****
3889 *****
3890 *****
3891 *****
3892 *****
3893 *****
3894 *****
3895 *****
3896 *****
3897 *****
3898 *****
3899 *****
3900 *****

```

Procedure: SF5YMP

Called by: SFREPORT

Calls: CASECONVERT
GETO
DICTREPORT()
DP.PRG

Uses: SF600TEX.DBF
SECTION.DBF
REPORT.DBF

Indexes: SF600TEX.IDX
SECTION.IDX
REPORT.IDX

FUNCTION sf5ymp


```

=> E mstr
4021 => ALEN(mdiag) msize
4022 => 10M mdiagname msize
4023 ""
4024 ""
4025 e = CHR(13)+CHR(10)
4026 => (mdagnosis)
4027 => diseasename(mdiagname)
4028 newline + newline + "expressions: " + IF( PROPER(mtemp))
4029 => en(mdiag) > 0
4030
4031 => getrulebase(marea, actthreshold, actthreshpro, actthreshlik)
4032 => strdiag(mdiagname, mdiag, mconf, cthreshold, m, is conf)
4033 mstr = mstr + newline + "Computer's suggested diagnoses: "
4034 FOR i = 1 TO ALEN(mdiagname)
4035 IF TYPE("mdiname(i)")="C"
4036 mstr = mstr + newline + trim(proper(mdiagname(i)))
4037 ELSE
4038 EXIT
4039 ENDIF
4040 ENDFOR
4041 mstr
4042
4043
4044 => rt treatment
4045 => rammer: HLL
4046
4047
4048
4049
4050
4051 Procedure: SFREPORT
4052 Called by: SFREPORT (procedure in DEMO.PRG)
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115

```

```

4059 => (mtreatment)
4060 => newline + newline + "plan:"
4061 mstr + newline + mtreatment
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115

```



```

1 *****
2 *****
3 *****
4 *****
5 *****
6 *****
7 *****
8 *****
9 *****
10 *****
11 *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****

Procedure file: C:\CAMD\WORK\DP.PRG

System: COMPUTER ASSISTED MEDICAL DIAGNOSIS
Author: ROY DOBBINS and HOA LY
Copyright (c) 1992 NAVAL HEALTH RESEARCH CENTER
Last modified: 08/04/92 15:10

Set by: WP2.PRG
      : WPAT (procedure in DEMO.PRG)
      : SFSYMP (procedure in DEMO.PRG)
      : WGETPATLIST (procedure in U.PRG)
      : SEARCH (procedure in U.PRG)
      : MORESEARCH (procedure in U.PRG)
      : GETPATDATA (procedure in U.PRG)
      : SAMSDATA (procedure in U.PRG)
      : SAVREC (procedure in U.PRG)
      : SFGGETPAT (procedure in U.PRG)
      : WGETPATLIST (procedure in U.PRG)
      : FILEFIND (procedure in U.PRG)
      : RESULT (procedure in WPATS.PRG)
      : PATEDIT (procedure in WPATS.PRG)

*****
***** Documented 10/16/92 at 10:52 *****
***** FoxDoc version 2.10 *****
*****
38 * Date: 08/04/92 10:01:15
39 * Program Name: Question.prg
40 * Author's Name: Hoa Le Ly
41 *
42 * Copyright (c) 1992 Company Name: NHRC
43 * Department: Code 22
44 * San Diego, CA 92138 - 5122
45 * Description: This program emulate $PIECE of MUMPS function. Which
46 * return the
47 * the portion of string which is bounded by the characters in deli
48 * miter. If both
49 * expr and expr2 are present, the value returned includes all char
50 * from the expr1-1th occurrence of delimiter, up to but not including
51 * the expr2th
52 * occurrence of delimiter. If expr2 is not present. It is assumed
53 * to have the same
54 * value as expr. If expr1 is not present. Then its value is assu
55 * me to be 1
56 * SYNTAX: DP(string, delimiter[,expr[,expr2]])
57 * PARAMETER: string: Character expression which character extract fro
58 * m
59 * delimiter: Character to delimiter
60 * expr: start of number occurrence of delimiter
61 * expr2: number of occurrence delimiter
62 * eg: string = "last, first age date of birth"
63 * 7dp(string,"",1) ==> last, first
64 * 7dp(string,"",1,2) ==> last, first age
65 *
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****
102 *****
103 *****

```

```

58 mp2 = -1
59 CASE PARAMETER() = 3
60 IF TYPE("mp") != "N"
61 RETURN ""
62 ELSE
63 IF mp < 1
64 RETURN ""
65 ENDIF
66 ELSE
67 mp2 = -1
68 CASE PARAMETER() = 4
69 IF TYPE("mp") != "N"
70 RETURN ""
71 ENDIF
72 IF TYPE("mp2") = "N"
73 mp2 = IIF(mp >= mp2, -1, mp2)
74 ELSE
75 mp2 = -1
76 ENDIF
77 ENDCASE
78 IF TYPE("ms") != "C"
79 ms = STR(ms)
80 ENDIF
81 IF OCCURS = OCCURS(mm,ms)
82 IF mp = 1
83 msr = IIF(mp = 1, ms, "")
84 RETURN msr
85 ELSE
86 mbegin = IIF(mp = 1, 1, (AT(mm,ms,(mp-1))+1))
87 IF mp2 = -1
88 mrend = AT(mm,ms,mp) - 1
89 IF mrend < 0
90 IF mbegin > 1
91 mrend = LEN(ms)
92 ELSE
93 RETURN ""
94 ENDIF
95 ELSE
96 mrend = AT(mm,ms,mp2) - 1
97 mrend = IIF(mrend < 0, LEN(ms), mrend)
98 IF
99 msr = SUBSTR(ms, mbegin, (mrend - mbegin + 1))
100 RETURN msr
101 *: EOF: DP.ACT
102
103

```

```

1 *****
2 Procedure file: C:\CAMD\WORK\ERRMSG.PRG
3
4 System: COMPUTER ASSISTED MEDICAL DIAGNOSIS
5 Author: ROY DOBBINS and HOA LY
6 Copyright (c) 1992, NAVAL HEALTH RESEARCH CENTER
7 Last modified: 07/30/92 16:40
8
9 Set by: GETPAD.PRG
10 : BROUSEPAT (procedure in U.PRG)
11 : ERNMSG (procedure in U.PRG)
12 : FILEMISSING (procedure in SECHECK.PRG)
13
14 Documented 10/14/92 at 10:52 FoxDoc version 2.10
15 *****
16 * Open a message window
17 *
18 * ERNMSG( <exp>[, <exp2>] )
19 *
20 PARAMETERS errmsg, timeout
21 PRIVATE ALL
22
23 * If no errmsg is sent then quit
24 IF PARAMETERS()=0
25 RETURN
26 ENDIF
27
28 * m.timeout is used in a WAIT TIMEOUT command
29 * to control the amount of time ERRMSG display's its message
30 * If no time limit is received set time to 0 wait forever
31
32 IF PARAMETERS()=1 OR EMPTY(m.timeout)
33 m.timeout=0
34 ENDIF
35
36 IF TYPE(m.timeout)='M'
37 m.timeout=INT(TYPE(m.timeout)*60)+INT(VAL(m.timeout)),0)
38 ENDIF
39
40 m.errmsg=IF(TYPE(m.errmsg)='C',"",m.errmsg)
41
42 IF PARAMETERS()=2 AND EMPTY(m.errmsg) AND m.timeout>0
43 RETURN
44 ENDIF
45
46 * set talk off
47 IF SET('TALK')='ON'
48 SET TALK OFF
49 savetalk='ON'
50 ELSE
51 savetalk='Off'
52 ENDIF
53
54 * Get the length of the message to size window
55 m.len=LEN(m.errmsg)
56
57 * Set minimum length for the Press any key
58 IF m.timeout=0
59 m.len=IF(m.len<28,28,m.len)
60 ENDIF
61
62 IF m.len>70
63 m.errmsg=SUBSTR(m.errmsg,1,70)
64 m.len=70
65 ENDIF
66

```

```

67 * find begining/ending of the window
68 m.begin=40-(INT(m.len/2)+1)
69 m.end=40+(INT(m.len/2)+1)
70
71 * Remember the current window status
72 m.oldwindow=IF( VOUTPUT()=errmsg, "", VOUTPUT() )
73
74 * Create the window
75 DEFINE WINDOW errmsg FROM 20,m.begin TO 23,m.end COLOR SCHEME 7
76 ACTIVATE WINDOW errmsg
77
78 * Print the message centered in the window
79
80 CLEAR
81 @ 0, ( WCOL() - LEN( m.errmsg ) )/2 SAY m.errmsg
82
83 IF m.timeout=0
84 m.preskey="Press any key to continue"
85 @ 1, ( WCOL() - LEN( m.preskey ) )/2 SAY m.preskey
86 ENDIF
87
88 * Wait for the number of seconds in m.timeout
89 * A value of 0 will wait forever
90 WAIT == TIMEOUT m.timeout
91
92 * Close Window
93 RELEASE WINDOW errmsg
94
95 * If there was no output window originally
96 IF EMPTY( m.oldwindow )
97
98 * Send future output back to the screen
99 ACTIVATE SCREEN
100
101 ELSE
102
103 * Return output to the original window
104 ACTIVATE WINDOW ( m.oldwindow )
105 ENDIF
106
107 SET TALK &savetalk
108 RETURN
109 * EOF: ERRMSG.ACT
110

```

```

62 @ 2,col SAY ""
63 ELSE
64 @ 2,32 SAY SPACE(1)
65 ENDOF
66 tpasswd = SUBSTR(tpasswd,1,LEN(tpasswd) - 1)
67 LOOP
68 ENDOF
69 ENDOF
70 IF (LASTKEY() = 27) OR (LASTKEY() = 5) ) 22 ESC
71 mflag = .f.
72 EXIT
73 ENDOF
74 IF (key1 = 13)
75 IF UPPER(tpasswd) == UPPER(mpass)
76 msuccess = .t.
77 mflag = .f.
78 ELSE
79 msuccess = .f.
80 mtry = mtry + 1
81 IF mtry > 2
82 mflag = .f.
83 =errmsg("Invalid User, Try next time !!!!!",1)
84 ELSE
85 =errmsg("Invalid Password, Try again !",1)
86 @ 2,33 GET temp_pass
87 CLEAR GETS
88 @ 2,32 SAY SPACE(1)
89 tpasswd = []
90 tcol = 32
91 ENDOF
92 ENDOF
93 ELSE
94 tpasswd = tpasswd + CHR(key1)
95 @ 2,col SAY [" "] COLOR SCHEME 10
96 ENDOF
97 tcol = tcol + 1
98 ENDDO
99 DEACTIVATE WINDOW get_pwd
100 RELEASE WINDOW get_pwd
101 SET ESCAPE timesc
102 SET TALK &talk
103 RETURN msuccess
104 *: EOF: GETPAD.ACT
105

```

```

*****
1  Procedure file: C:\CAMD\WORK\KEYLOG.PRG
2
3
4
5  System: COMPUTER ASSISTED MEDICAL DIAGNOSIS
6  Author: ROY DOBBINS and HOA LY
7  Copyright (c) 1992 NAVAL HEALTH RESEARCH CENTER
8  Last modified: 08/03/92 15:46
9
10 Set by: M PAT (procedure in DEMO.PRG)
11 : M_WELNC (procedure in DEMO.PRG)
12 : M_TRAINING (procedure in DEMO.PRG)
13 : BACKUP (procedure in DEMO.PRG)
14 : RESTORE (procedure in DEMO.PRG)
15 : M_LOOKUP (procedure in DEMO.PRG)
16 : M_DATA (procedure in DEMO.PRG)
17
18 Calls: FILEFIND (procedure in U.PRG)
19
20 Uses: KEYLOG.DBF
21
22 Indexes: KEY.IDX
23 : KEYDATE.IDX
24
25 Documented 10/14/92 at 10:52 FoxDoc version 2.10
26 *****
27 * keylog.prg
28 *
29 *
30 * Computer Assist Medical Diagnosis System
31 *
32 * Revision: 1.0 Date: 92/08/03 11:11:10
33 * Author: Hoa L. LY
34 *
35 *
36 * This procedure store every action to keylog.dbf.
37 *
38 * Keylog.dbf has index on action and date
39 *
40
41 PARAMETER maction
42 PRIVATE mrfind, msel
43 msel = SELECT()
44 mrfind = filefind('keylog.dbf')
45 IF mrfind
46 RETURN .F.
47 ENDIF
48 IF PARAMETER() > 0
49 SELECT 0
50 USE keylog AGAIN INDEX KEY, keydate
51 APPEND BLANK
52 REPLACE ACTION WITH maction
53 REPLACE DATE WITH DATE()
54 REPLACE TIME WITH TIME()
55 USE
56 ENDIF
57 SELECT (msel)
58 RETURN
59 *: EOF: KEYLOG.ACT

```

```

1 *****
2 ** Procedure file: C:\CAMD\WORK\PADEDT.PRg
3 **
4 **
5 ** System: COMPUTER ASSISTED MEDICAL DIAGNOSIS
6 ** Author: ROY DOBBINS and HOA LY
7 ** Copyright (c) 1992, NAVAL HEALTH RESEARCH CENTER
8 ** Last modified: 07/29/92 9:25
9 **
10 ** Set by: GPOPPFLOY (procedure in CAMD.MPR)
11 ** : CAMDTR.MPR
12 ** Uses: SYSTEM.DBF
13 **
14 **
15 ** Documented 10/14/92 at 10:52 FoxDoc version 2.10
16 ** *****
17 **
18 ** * Validation the password. Is boolean returnable
19 **
20 **
21 PRIVATE ALL
22 IF SET("TALK") == "ON"
23 mtlalk = "ON"
24 ELSE
25 mtlalk = "OFF"
26 ENDIF
27 SET TALK OFF
28 startdir=FULLPATH(SYS(2003))
29 mpsdata = FULLPATH(GETENV("CAMD"))
30 mopath = SET("PATH")
31 mpath = startdir + ";" + mpsdata
32 ** files
33 SET PATH TO (mpath)
34 SELECT 0
35 USE SYSTEM AGAIN
36 mflag = ".f."
37 mesc = SET("ESCAPE")
38 SET ESCAPE OFF
39 DEFINE WINDOW get_pwd FROM 10,1' TO 16,63 DOUBLE FLOAT NOCLOSE GROW 2
40 ** COLOR SCHEME 5
41 ACTIVATE WINDOW get_pwd
42 temp_pass = SPACE(8)
43 @ 2,7 SAY "ENTER NEW PASSWORD: "
44 CLEAR GETS
45 @ 2,33 GET temp_pass
46 tpasswr=()
47 tcol = 33
48 msuccess = .f.
49 mtry = 0
50 DO WHILE mflag
51 key1 = 0
52 DO WHILE key1 = 0
53 key1 = INKEY()
54 IF ( (LASTKEY() >= -9 .AND. LASTKEY() <= -1) OR (LASTKEY() = 16) )
55 ignore F2 to F10 and DEL key
56 key1 = 0
57 LOOP
58 ENDIF
59 IF LASTKEY() = 28 ** F1
60 **HELP PASSWORD
61 key1 = 0
62 LOOP

```

```

63 ENDIF
64 IF ( (LASTKEY() = 19) OR (LASTKEY() = 127) ) ** back arrow or be
65 ** ck space
66 IF tcol = 33
67 key1 = 0
68 LOOP
69 ELSE
70 tcol = tcol - 1
71 @ 2,tcol SAY [ ] COLOR SCHEME 10
72 IF tcol > 33
73 @ 2,(tcol-1) SAY [ ] COLOR SCHEME 10
74 ELSE
75 @ 2,32 SAY SPACE(1)
76 ENDIF
77 tpasswr = SUBSTR(tpasswr,1,LEN(tpasswr) - 1)
78 LOOP
79 ENDIF
80 IF ( (LASTKEY() = 27) OR (LASTKEY() = 5) ) ** ESC
81 mflag = .f.
82 EXIT
83 ENDIF
84 IF (key1 = 13)
85 IF EMPTY(tpasswr) AND LEN(tpasswr) < 10
86 mpass = UPPER(tpasswr)
87 REPLACE password WITH mpass
88 mflag = .f.
89 ENDIF
90 ELSE
91 tpasswr = tpasswr + CHR(key1)
92 @ 2,tcol SAY [ ] COLOR SCHEME 10
93 ENDIF
94 tcol = tcol + 1
95 ENDDO
96 DEACTIVATE WINDOW get_pwd
97 RELEASE WINDOW get_pwd
98 USE
99 SET ESCAPE &mesc
100 SET TALK &mtalk
101 RETURN msuccess
102 ** EOF: PADEDT.ACT

```



```

133 #REGION 1
134 mfilename = mprintfile
135 RETURN .T.
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150

```

_q710LH94d mbuttons VALID

Function Origin:

From Screen: 6
Variable: u2, mbuttons
Called By: VALID Clause
Object Type: Push Button
Snippet Number: 2

```

151 FUNCTION _q710LH94d 22 mbuttons VALID
152 IF mbuttons = 2
153     mprintfile = ""
154 ELSE
155     mprintfile = mfilename
156 ENDIF
157 RETURN
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172

```

_q710LH94q mchoice VALID

Function Origin:

From Screen: 7
Variable: u2, mchoice
Called By: VALID Clause
Object Type: Radio Button
Snippet Number: 3

```

173 FUNCTION _q710LH94q 22 mchoice VALID
174 #REGION 1
175 IF mchoice = 1
176     mprintfile = mfilename
177     CURSORJ = OBJNUM(mprintfile)
178 ELSE
179     mprintfile = "PRINT"
180 ENDIF
181 SHOW GETS
182 mfilename = mprintfile
183 RETURN .T.
184 *: EOF: 02.ACT
185
186

```

```

1 * Procedure file: C:\CAMD\WORK\QUESTION.PRG
2 *
3 *
4 *
5 * System: COMPUTER ASSISTED MEDICAL DIAGNOSIS
6 * Author: ROY DOBBINS and NOA LY
7 * Copyright (c) 1992 NAVAL HEALTH RESEARCH CENTER
8 * Last modified: 08/04/92 14:48
9 *
10 * Procs & Fncts: PRINTQUES()
11 * : WRITE
12 * : OPENFILE
13 *
14 * Set by: M_DATA (procedure in DEMO.PRG)
15 *
16 * Calls: Q2.PRG (function in QUESTION.PRG)
17 * : PRINTQUES()
18 *
19 * Documented 10/14/92 at 10:52 FoxDoc version 2.10
20 *
21 * *****
22 * Date: 08/04/92 10:01:15
23 * Program Name: Question.prg
24 * Author's Name: Noe Le Ly
25 *
26 * Copyright (c) 1992 Company Name: MHRC
27 * Department: Code 22
28 * San Diego, CA 92138 - 5122
29 * Description: This program print out the list of question for each
30 * area input
31 * PARAMETER: area id (numeric type)
32 * *****
33
34 * PARAMETER area
35 * PUBLIC mprIntfile
36 *
37 * mprIntfile = ""
38 * mbuttons = 0
39 * mudf = SET("DIFPARMS")
40 * SET LOFPARMS TO REFERENCE
41 * mprIntfile = q2(mprIntfile)
42 *
43 * IF mbuttons = 2
44 * RETURN
45 * ENDIF
46 * SET LOFPARMS TO mudf
47 *
48 * =printques(mprIntfile)
49 *
50 * CLOSE DATABASE
51 * RETURN
52 *
53 *
54 * write the question and it's alternate of reponse
55 *
56 *
57 *
58 *
59 * Procedure: WRITE
60 * Called by: PRINTQUES() (function in QUESTION.PRG)
61 * Calls: OPENFILE (procedure in QUESTION.PRG)
62 *
63 *
64 *
65 *
66 *
67 *
68 *
69 *
70 *
71 *
72 *
73 *

```

```

74 *****
75 FUNCTION write
76 PRIVATE msel
77
78 *openfile()
79
80 SELECT area
81 SEEK (marea)
82 IF FOUND()
83 ? PADC(UPPER(name),80)
84 ?
85 ENDIF
86
87 SELECT area
88 SET RELATION TO area INTO rule
89 SET RELATION TO id INTO enum, id INTO dictval
90 SET RELATION TO id INTO enum, id INTO dictval
91 SET FILTER TO area = marea
92 SET RELATION TO premise INTO premise, action INTO action
93
94 GO TOP
95 pid = 0
96
97 DO WHILE IEOF()
98 IF premise.id <> pid
99 mid = premise.id
100 SELECT dict
101 IF dict.askable
102 ? ALLTRIM(question)
103 DO CASE
104 CASE dict.datatype = "N"
105 SELECT enum
106 DO WHILE id = mid
107 ? SPACE(10) + ALLTRIM(enumerate)
108 SKIP
109 CASE dict.datatype = "M"
110 SELECT dictval
111 ? SPACE(10) + "Range" + SPACE(1) + ALLTRIM(STR(id)) + " t"
112 ALLTRIM(STR(hi)) + SPACE(1) + units
113 ENDCASE
114 ENDOF
115 pid = mid
116 ENDOF
117 SELECT rule
118 SKIP
119 ENDOF
120 RETURN
121
122 *
123 *
124 * open rule file
125 *
126 *
127 *****
128 Procedure: OPENFILE
129 Called by: WRITE (procedure in QUESTION.PRG)
130 Uses: AREA.DBF
131 : DICT.DBF
132 : VAL.DBF
133 : ENUM.DBF
134

```

```

195 SET PRINTER TO
196 SET CONSOLE ON
197 ENDIF
198 RETURN .T.
199
201 *: EOF: QUESTION.ACT

```

```

155 *) PREMISE.DBF
156 *) ACTION.DBF
157 *) RULE.DBF
158
159 Indexes: AREA.IDX
160 *) DICTID.IDX
161 *) DICTNAME.IDX
162 *) VAL.IDX
163 *) ENLN.IDX
164 *) PREMISE.IDX
165 *) ACTION.IDX
166 *) SALIENCE.IDX
167 *) RULE.IDX
168
169 *****
170 FUNCTION openfile
171
172 SELECT 0
173 USE area AGAIN INDEX area
174 SELECT 0
175 USE dict AGAIN INDEX dictid,dictname
176 SELECT 0
177 USE VAL AGAIN INDEX VAL ALIAS dictval
178 SELECT 0
179 USE enln AGAIN INDEX enln
180 SELECT 0
181 USE premise INDEX premise
182 SET RELATION TO id INTO dict
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

10/09/92          RESTORE.SPR          14:10:03

Author's Name

Copyright (c) 1992 Company Name
Address
City,           Zip

Description:
This program was automatically generated by GENSCRM.

RESTORE Setup Code - SECTION 1

REGION 1
PARAMETERS MESSAGE, wildcard, filename
PRIVATE mfname, mpath, olddrive, predrive, oldpath, maction, mdrive,
>> mfiles
--DO CASE
--CASE PARAMETERS() = 0
  m.message = ""
  m.wildcard = "*.zip"
  mfname = ""
--CASE PARAMETERS() = 1 OR EMPTY(m.wildcard)
  m.wildcard = "*.zip"
  mfname = ""
--CASE PARAMETERS() = 2 OR EMPTY(m.filename)
  mfname = ""
--OTHERWISE
  mfname = filename
--ENDCASE

REGION 0
REGIONAL m.curarea, m.talkstat, m.compatstat

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
--ELSE
  m.talkstat = "OFF"
--ENDIF
m.compatstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

Window definitions

IF NOT MEXIST("w.restore")
  DEFINE WINDOW "w.restore" ;
    FROM INT((SCOL()-16)/2),INT((SCOL()-61)/2) ;

```

```

66 TO INT((SROW()-18)/2)+17,INT((SCOL()-61)/2)+60 ;
67 NOFLOAT ;
68 NOCLOSE ;
69 SHADOW ;
70 DOUBLE ;
71 COLOR SCHEME 5
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
106
```

```

132  DEFAULT 1;
133  VALID cprbud388();
134  COLOR SCHEME 5, 6;
135  @ 9,45 GET maction;
136  PICTURE "a-vt \\\-restore;\-Cancel";
137  SIZE 1,9,1;
138  DEFAULT 1;
139  VALID cprbud37();
140  @ 2,1 GET mfile;
141  PICTURE "a-vt \\\-restore;\-Cancel";
142  FROM filearray;
143  SIZE 11,26;
144  DEFAULT 1;
145  VALID cprbud32();
146  COLOR SCHEME 6;
147  @ 14,19 GET mfname;
148  SIZE 1,39;
149  DEFAULT 1;
150  VALID cprbud30();
151  @ 14,0 SAY "restore file name: ";
152  @ 3,28 SAY "from drive: ";
153  [IF NOT MWISIBLE("restore")
154  ACTIVATE WINDOW "w_restore"
155  ENDIF]
156
157  READ CYCLE MODAL
158  RELEASE WINDOW "w_restore"
159
160  #REGION 0
161  IF m.talkstat = "NON"
162  SET TALK ON
163  ENDIF
164  IF m.compstat = "NON"
165  SET COMPATIBLE ON
166  ENDIF
167
168  *
169  *
170  *
171  *
172  *
173  *
174  *
175  *
176  *
177  #REGION 1
178  SET DEFAULT TO (m.olddrive + m.oldpath)
179  RETURN
180
181  *****
182  *****
183  *****
184  *****
185  *****
186  *****
187  *****
188  *****
189  *****
190  *****
191  *****
192  *****
193  *****
194  *****
195  *****
196  *****
197  *****

```

RESTORE Cleanup Code

```

198  m.olddrive = ON( "ERROR" )
199  m.error = .F.
200  ON ERROR m.error = .T.
201
202  * Create the array of legal drive names
203  DECLARE drivearray[ 1 ] = ""
204  drivearray[ 1 ] = ""
205
206  * Loop from A to Z
207  m.driveName = "A"
208  FOR i = 1 TO 26
209
210  * Try to switch to the drive
211  SET DEFAULT TO (m.driveName)
212
213  * If it worked
214  IF NOT m.error
215
216  * Add the name to the last element in the array
217  drivearray[ ALEN( DriveArray ) ] = m.driveName + " "
218
219  * Add another element at the end of the array
220  DECLARE drivearray[ ALEN( DriveArray ) + 1 ]
221  ENDIF
222
223  * Change to the next letter in the alphabet
224  m.driveName = CHR( ASC( m.driveName ) + 1 )
225
226  * Reset the error trap
227  m.error = .F.
228  NEXT
229
230  * If the first element is empty, no drives were found
231  IF EMPTY( drivearray[ 1 ] )
232  SHOW GET m.drive disabled
233  ELSE
234  * Drives were found so cut off the last empty element
235  * added to the array in the loop
236  DECLARE drivearray[ ALEN( DriveArray ) - 1 ]
237  ENDIF
238
239  * Reset the error trap and return to home
240  ON ERROR &olddrive
241  SET DEFAULT TO (m.olddrive)
242
243  * Initialize the popup variable to the element in the
244  * drive array which contains the current drive
245  mdrive = ASCAN( drivearray, m.olddrive )
246  RETURN
247
248  *****
249  *****
250  *****
251  *****
252  *****
253  *****
254  *****
255  *****
256  *****
257  *****
258  *****
259  *****
260  *****
261  *****
262  *****
263  *****

```

```

264 m.dirstring = CURDIR()
265
266 * If we are not in the root directory
267 -IF LEN( m.dirstring ) > 1
268
269 * Start parsing the path string for each directory name
270 m.dircount = 1
271
272 * Continue as long as there is a pair of slashes
273 * Surrounding the next part of the string
274 -DO WHILE AT( "\", m.dirstring, m.dircount ) <> 0 AND ;
275 AT( "\", m.dirstring, m.dircount + 1 ) <> 0
276
277 * Add another element at the end of the array
278 DECLARE patharray[ m.dircount + 1 ]
279
280 * Cut out the next set of characters between the slashes
281 m.firstchar = AT( "\", m.dirstring, m.dircount ) + 1
282
283 m.pathlength = AT( "\", m.dirstring, m.dircount + 1 ) ;
284 - m.firstchar
285
286 * And add them to the next array element
287 patharray[ m.dircount + 1 ] = SUBSTR( m.dirstring, ;
288 m.firstchar, m.pathlength )
289
290 * Look for the next directory name in the path string
291 m.dircount = m.dircount + 1
292
293 -ENDDO
294
295 * Point the popup variable at the last element in the array
296 mpath = ALEN( patharray )
297
298 RETURN
299
300 *****
301 PROCEDURE newfilepop
302 *****
303
304 * Create a popup with all of the file names and its subdirection in t
305 => he current directory
306 * This will be based on an array as well
307
308 * Create an array with all the files matching the current wild card
309 * ADIR() creates an array with 5 columns.
310
311 PRIVATE startsort
312 * Fill an array with directory names only
313 =ADIR( dirarray, "", "d")
314 SIZE = ALEN( dirarray, 1 )
315
316 -IF dirarray[ 1, 1 ] = "."
317 =ADEL( dirarray, 1 )
318 SIZE = SIZE - 1
319 DECLARE dirarray[ size, 5 ]
320
321 * We must be in the root directory "\"
322 * Add one more row to the directory array
323 SIZE = SIZE + 1
324 DECLARE dirarray[ size, 5 ]
325
326 * Push all the rows down by one
327 =AINS( dirarray, 1 )
328
329 * Fill in the first directory name in the array
330 * a bug in the popups makes it refuse to display a

```

RESTORE.AC1 10-14-92 10:52a

```

329 * prompt of "\", use "\" to make one \ appear
330 * even then the bar will automatically be non-selectable
331 * which in this case is fine
332 dirarray[ 1, 1 ] = "\"
333
334 * Start the sort after the root name
335 -ENDIF
336
337 -IF ALEN( dirarray, 1 ) > 2
338
339 * Sort the array starting at the starting row
340 =ASORT( dirarray, AELEMENT( dirarray, 2, 1 ) )
341
342 -ENDIF
343
344 FOR i = 1 TO ALEN( dirarray, 1 )
345 dirarray[ i, 1 ] = "[" + dirarray[ i, 1 ] + "]"
346
347 -ENDFOR
348
349 -IF ADIR( filearray, m.wildcard ) = 0
350
351 SIZE = ALEN( dirarray, 1 )
352 DECLARE filearray[ size + 1, 5 ]
353 =ACOPY( dirarray, filearray )
354
355 -ELSE
356 stop = ALEN( dirarray, 1 )
357 FOR i = 1 TO stop
358 SIZE = ALEN( filearray, 1 )
359 DECLARE filearray[ size + 1, 5 ]
360 =AINS( filearray, 1 )
361 filearray[ i, 1 ] = dirarray[ ALEN( dirarray, 1 ), 1 ]
362 IF ALEN( dirarray, 1 ) > 1
363 DECLARE dirarray[ ALEN( dirarray, 1 ) - 1, 5 ]
364
365 -ENDIF
366 -ENDIF
367
368 mfile = 1
369 RETURN
370
371 *****
372 FUNCTION pathstring
373 *****
374 * Convert an array of subdirectories, such as PathArray,
375 * into a legal path string for use with SET DEFAULT
376 * Use the current value of the path popup as the
377 * ending point on the path
378 PRIVATE ppath
379
380 * Start with the drive name
381 m.ppath = patharray[ 1, 1 ]
382
383 * If the path popup is pointing to a subdirectory
384 -IF mpath > 1
385
386 * Add all the subdirectories to the path string
387 FOR i = 2 TO mpath
388 m.ppath = m.ppath + "\" + patharray[ i, 1 ]
389
390 -NEXT
391
392 * End the path with one last slash for good luck
393 m.ppath = m.ppath + "\"
394
395 RETURN m.ppath
396
397 *****
398 PROCEDURE pkmzip
399 *****

```

Page 3 of 5

```

395 PRIVATE CURDIR, datadir, msel
396 IF EMPTY(GETENV("CAMOUTIL"))
397 DO setpath WITH GETENV("CAMOUTIL")
398 ENDIF
399 CURDIR = FULLPATH(CURDIR())
400 IF EMPTY(GETENV("CAM"))
401 datadir = FULLPATH(GETENV("CAM"))
402 ELSE
403 datadir = CURDIR()
404 ENDIF
405 desdir = pathstring() + mfname
406 SET DEFA TO (datadir)
407 msel = SELECT()
408 SELECT 0
409 USE backup AGAIN
410 COPY TO t0000000.txt TYPE SDF
411 IF EMPTY(GETENV("CAMOUTIL"))
412 pkzipcom = "j" + GETENV("CAMOUTIL") + "\PKUNZIP -O &desdir &curdir &t0000000
413 => 00.txt"
414 ELSE
415 pkzipcom = "PKUNZIP -O &desdir &t0000000.txt"
416 ENDIF
417 &pkzipcom
418 USE
419 DELETE FILE t0000000.txt
420 SET DEFA TO (CURDIR)
421 RETURN
422 *
423 *
424 *
425 *
426 *
427 *
428 *
429 *
430 *
431 *
432 *
433 *
434 *
435 *
436 *
437 FUNCTION _qR0ud7na 22 mdrive WHEN
438 #REGION 1
439 m.prevdribe = mdrive
440 *
441 *
442 *
443 *
444 *
445 *
446 *
447 *
448 *
449 *
450 *
451 *
452 *
453 *
454 *
455 *
456 *
457 *
458 *
459 *

```

_qR0ud7na mdrive WHEN

Function Origin:

From Screen: RESTORE, Record Number: 2

Variable: mdrive

Called By: WHEN Clause

Object Type: Popup

Snippet Number: 1

_qR0ud7u5 mdrive VALID

Function Origin:

From Screen: RESTORE, Record Number: 2

Variable: mdrive

Called By: VALID Clause

Object Type: Popup

Snippet Number: 2

```

455 *
456 *
457 *
458 *
459 *

```

```

460 *
461 *
462 *
463 *
464 *
465 *
466 *
467 *
468 *
469 *
470 *
471 *
472 *
473 *
474 *
475 *
476 *
477 *
478 *
479 *
480 *
481 *
482 *
483 *
484 *
485 *
486 *
487 *
488 *
489 *
490 *
491 *
492 *
493 *
494 *
495 *
496 *
497 *
498 *
499 *
500 *
501 *
502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *

```

_qR0ud888 mpath VALID

Function Origin:

From Screen: RESTORE, Record Number: 5

Variable: mpath

Called By: VALID Clause

Object Type: Popup

Snippet Number: 3

_qR0ud8c7 maction VALID

Function Origin:

From Screen: RESTORE, Record Number: 6

Variable: maction

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 4

```

502 *
503 *
504 *
505 *
506 *
507 *
508 *
509 *
510 *
511 *
512 *
513 *
514 *
515 *
516 *
517 *
518 *
519 *
520 *
521 *
522 *
523 *
524 *

```



```

525 *
526 *
527 *
528 *
529 *
530 *
531 *
532 *
533 *
534 *
535 *
536 *
537 *
538 *
539 *
540 FUNCTION _qPr0ud8p2 22 mfile VALID
541 #REGION 1
542 mnewfile = filearray(mfile,1)
543 IF "(" $ mnewfile
544   mnewpath = SUBSTR(mnewfile,2,LEN(mnewfile)-2)
545   SET DEFA TO (mnewpath)
546   DO newpathpop
547   DO newfilepop
548   SHOW GETS
549 ELSE
550   mfname = mnewfile
551   SHOW GETS
552 ENDIF
553 *
554 *
555 *
556 *
557 *
558 *
559 *
560 *
561 *
562 *
563 *
564 *
565 *
566 *
567 *
568 FUNCTION _qPr0ud8zo 22 mfname VALID
569 #REGION 1
570 IF EMPTY(mfname)
571   IF IFILE(mfname)
572     mfname = ""
573   ENDIF
574 ENDIF
575 *: EOF: RESTORE.AC1
576

```

_qPr0ud8p2 mfile VALID

Function Origin:

From Screen: RESTORE, Record Number: 7

Variable: mfile

Called By: VALID Clause

Object Type: List

Snippet Number: 5

_qPr0ud8zo mfname VALID

Function Origin:

From Screen: RESTORE, Record Number: 8

Variable: mfname

Called By: VALID Clause

Object Type: Field

Snippet Number: 6

```

1 *****
2 **
3 ** Procedure file: C:\CAMD\WORK\SECHECK.PRG
4 **
5 ** System: COMPUTER ASSISTED MEDICAL DIAGNOSIS
6 ** Author: ROY DOBBINS and MOA LY
7 ** Copyright (c) 1992, NAVAL HEALTH RESEARCH CENTER
8 ** Last modified: 09/03/92 12:01
9 **
10 ** Procs & Fncts: GETSYSCODE()
11 ** : SYSWARN()
12 ** : SYSCLAIM()
13 ** : FILEMISSING
14 **
15 ** Set by: DEMO.PRG
16 **
17 ** Calls: GETSYSCODE() (function in SECHECK.PRG)
18 ** : SYSWARN() (function in SECHECK.PRG)
19 ** : GETPMO.PRG
20 ** : BIGCHARS.PRG
21 ** : SYSCLAIM() (function in SECHECK.PRG)
22 **
23 ** Documented 10/14/92 at 10:52 FoxDoc version 2.10
24 ** *****
25 **
26 ** * Security check for user
27 ** *****
28 ** PUBLIC msyswarning, mprivateact, mdiscclaim, mpassword
29 ** PRIVATE moldpath, mpath
30 ** IF SET("TALK")="ON"
31 ** SET TALK OFF
32 ** mtalk = "ON"
33 ** ELSE
34 ** mtalk = "OFF"
35 ** ENDOF
36 **
37 ** startdir=FULLPATH(SYS(2003))
38 ** mpsdata = FULLPATH(GETENV("CAMD"))
39 ** moldpath = SET("PATH")
40 ** mpath = startdir + ";" + mpsdata
41 **
42 ** SET PATH TO (mpath)
43 ** =getsyscode(amsyswarning, amprivateact, amdiscclaim, ampassword)
44 ** msyswin=msyswarn(msyswarning)
45 ** muservalid = getpmo(mpassword)
46 ** RELEASE WINDOW amsyswin
47 ** IF muservalid
48 ** =bigchars(6,17,"CAMD".6)
49 ** =sysclain(mdiscclaim)
50 ** ENDOF
51 ** SET PATH TO (moldpath)
52 **
53 ** SET TALK &mtalk
54 ** RETURN muservalid
55 **
56 ** * Print warning text to screen
57 ** *****
58 **
59 ** function: SYSWARN()
60 ** *****

```

```

68 ** Called by: SECHECK.PRG
69 ** *****
70 **
71 ** FUNCTION syswarn
72 ** PARAMETER msyswar
73 ** IF PARAMETER() = 0
74 ** RETURN
75 ** ENDOF
76 ** mtext = msyswar
77 ** DEFINE WINDOW syswarn FROM 1,0 TO 6,76 NOFLOAT NOCLOSE COLOR SCHEME 7
78 ** ACTIVATE WINDOW syswarn SAME
79 ** @ 0,1 SAY mtext ;
80 ** SIZE 4,73
81 ** RETURN "syswarn"
82 ** *****
83 **
84 ** * Get system security text and password
85 ** *****
86 **
87 ** Function: GETSYSCODE()
88 ** *****
89 **
90 ** Called by: SECHECK.PRG
91 **
92 ** Calls: FILEMISSING (procedure in SECHECK.PRG)
93 **
94 ** Uses: SYSTEM.DBF
95 ** *****
96 **
97 ** FUNCTION getsyscode
98 ** PARAMETER mwarn, mprivate, mdisc, mpass
99 ** msel = SELECT()
100 ** IF NOT USED("SYSTEM")
101 ** IF NOT FILE("system.dbf")
102 ** ON ERROR DO filemissing
103 ** msel = LOCFILE("SYSTEM", "DBF", "Locate File SYSTEM.DBF")
104 ** ENDOF
105 ** ON ERROR
106 ** SELECT 0
107 ** USE SYSTEM
108 ** ELSE
109 ** SELECT SYSTEM
110 ** ENDOF
111 ** mwarn = syswarning && system warning text
112 ** mprivate = privacyact && system privacy act text
113 ** mdisc = sysdisc && system disc claim text
114 ** mpass = ALLTRIM(password) && system password
115 ** USE
116 ** SELECT (msel)
117 ** RETURN .T.
118 **
119 ** * Error for filemissing
120 ** *****
121 **
122 ** Procedure: FILEMISSING
123 ** *****
124 **
125 ** Called by: GETSYSCODE() (function in SECHECK.PRG)
126 **
127 ** Calls: ERRMSG.PRG
128 ** *****

```

```

132 *|*****
124 PROCEDURE filemissing
125   errmsg("file missing !!!!!!!")
126 ON ERROR
127   CANCEL
128   RETURN
129
130 *|*****
131 * System disclaimer
132 *|*****
133 *|*****
134 *|*****
135 *|*****
136 *|*****
137 *|*****
138 *|*****
139 *|*****
140 *|*****
141 *|*****
142 *|*****
143 *|*****
144 *|*****
145 *|*****
146 *|*****
147 *|*****
148 *|*****
149 *|*****
150 *|*****
151 *|*****
152 *|*****
153 *|*****
154 *|*****
155 *|*****
156 *|*****
157 *|*****
158 *|*****
159 *|*****
160 *|*****
161 *|*****
162 *|*****
163 *|*****
164 *|*****
165 *|*****
166 *|*****
167 *|*****
168 *|*****
169 *|*****
170 *|*****

```

Function: SYSCLAIM()

Called by: SECHECK.PRG

FUNCTION sysclaim

PARAMETER m.txt

IF NOT EXIST("w_claim")

DEFINE WINDOW w_claim

FROM INT((SROW()-22)/2),INT((SCOL()-79)/2) ;

TO INT((SROW()-22)/2)+1,INT((SCOL()-79)/2)+78 ;

NOFLOAT ;

NOCLOSE ;

SHADOW ;

DOUBLE ;

COLOR SCHEME 1

END IF

IF WVISIBLE("w_claim")

ACTIVATE WINDOW w_claim SAME

ELSE

ACTIVATE WINDOW w_claim NOSHOW

END IF

@ 0,0 EDIT m.txt ;

PICTURE "a1" ;

SIZE 18,77,0 ;

DEFAULT " " ;

NO MODIFY ;

SCROLL

ACTIVATE WINDOW w_claim

WAIT

CLEAR GETS

RELEASE WINDOW w_claim

RETURN

*: EOF: SECHECK.ACT

```

1 *****
2
3 Procedure file: C:\CAMD\WORKU.PRQ
4
5 System: COMPUTER ASSISTED MEDICAL DIAGNOSIS
6 Author: ROY DOBBINS and HOA LY
7 Copyright (c) 1992, NAVAL HEALTH RESEARCH CENTER
8 Last modified: 10/13/92 10:21
9
10 Procs & Fncts: USERINIT()
11 : POPUPSHOW()
12 : POPUPHIDE()
13 : OPENFILES()
14 : MAINMENU()
15 : USERCLOSE()
16 : GETEXAREA()
17 : DISEASENAH()
18 : ENCTITLE()
19 : WCONFIRM()
20 : PATID()
21 : FLUSHFILES()
22 : POPMESSAGE()
23 : DISEASEID()
24 : PATSSN()
25 : ADDENCT()
26 : NEWENCT()
27 : PUTENCT()
28 : PUTSYMPTOM()
29 : PUTDIAGNOS()
30 : NEXTENCT()
31 : GETRULEBASE()
32 : DISEASECNT()
33 : DISNAMES()
34 : GETDISLIST()
35 : AREACOUNT()
36 : AREANAMES()
37 : GETAREALIST()
38 : PATCOUNT()
39 : PATNAMES()
40 : GETPATLIST()
41 : ENCCOUNT()
42 : ENCNAMES()
43 : GETENCLIST()
44 : GETSTATION()
45 : ENGETINFO()
46 : GETDIAGNOS()
47 : GETSYMPTOM()
48 : UNDERMENU()
49 : PATGETINFO()
50 : ERRMESSAGE()
51 : ISVALIDSSN()
52 : PATPUTINFO()
53 : ADPAT()
54 : NEWPATIENT()
55 : DICTONAME()
56 : DICTCOUNT()
57 : DICTQ()
58 : DICTOV()
59 : EDNUM()
60 : EDBOOK()
61 : DICTECOUNT()
62 : DICTRE()
63 : EDCHECK()
64 : CHANGELIST()
65 : DICTREPORT()
66 : SPLIT()

```

```

67 : WCHOICE
68 : GETDISLOOKUPLIST
69 : EDLIST
70 : EDHECK
71 : SETCHECK
72 : _EVALID
73 : _EDITVOTAG
74 : _EDITCOMPL
75 : _EDITDIAG
76 : _EDITNAMOP
77 : _EDITTASSIS
78 : _EDITTREAT
79 : _RELEASE
80 : _PRINTENC
81 : _STOPREAD
82 : _RYHANDLER
83 : _HIT
84 : _QUIT
85 : _PVALID
86 : _PUSHHELP
87 : _POPHELP
88 : _SETHelp
89 : _OLMR
90 : _INITHELP
91 : _GETCOLORS
92 : _OPENDIAG
93 : _OPENDICT
94 : _CLOSEDICT
95 : _DICTORD
96 : _DICTENUM
97 : _DICTQID
98 : _OPENENCT
99 : _FLUSHENCT
100 : _ENCTID
101 : _DISEASEGET
102 : _DISEASEPUT
103 : _CLOSEFILES
104 : _OPENPAT
105 : _FLUSHPAT
106 : _DELPATIENT
107 : _TSVALIDPAT
108 : _STRNAME
109 : _OPENRULE
110 : _CLOSERULE
111 : _GETRULE
112 : _OPENSYP
113 : _GETOPTS
114 : _NEXTPAT
115 : _NEWENCT
116 : _NEXTUSER
117 : _ADDOUSER
118 : _WPGETPATLIST
119 : _SEARCH
120 : _MORESEARCH
121 : _GETCANDPAT
122 : _GETSAMPAT
123 : _GETPATDATA
124 : _GETSAMS
125 : _SAMSDATA
126 : _GETPAT
127 : _GETPATTEMP
128 : _PATDATA
129 : _COMBINED
130 : _SAVEC
131
132

```

```

133 * :
134 * :
135 * : L SECONVERT
136 * : SPENCOUNT
137 * : SPGETENC
138 * : SPENCLIST
139 * : SPGETPAT
140 * : SPENCNAMES
141 * : SPGETPATLIST
142 * : SPPRINT
143 * : SETPATH
144 * : ERRHAND
145 * : FILEFIND
146 * : REINDEX
147 * :
148 * : Documented 10/14/92 at 10:50 FoxDoc version 2.10
149 * : *****
150 * : u.prg
151 * :
152 * : Medical Practice Support System
153 * : $Revision: 1.1 $ $Date: 92/05/04 11:11:10 $
154 * : $Author: RoyDobbins $
155 * : 4/24/92 rwd02 context sensitive help settings
156 * : 4/27/92 rwd03 batch test mode
157 * : 5/14/92 rwd04 simple report format
158 * : 6/1/92 rwd06 on-line medical reference process
159 * : 6/2/92 rwd07 performance tuning of assist
160 * : 6/9/92 hll01 delete variable "startdir"
161 * : 6/10/92 hll02 add procedures and functions for patient intergrate
162 * : with SAMS
163 * : 6/10/92 hll03 add procedures and functions for doing sf600 report
164 * : 6/18/92 hll04 update patient info (sex and age) to working memory
165 * : 6/18/92 hll05 comment the computer diagnosis out from sf600 report
166 * : 6/18/92 hll06 add mreport to PUBLIC variable for getting the sf600
167 * : report
168 * :
169 * :
170 * :
171 * :
172 * :
173 * :
174 * :
175 * :
176 * :
177 * :
178 * :
179 * :
180 * :
181 * :
182 * :
183 * :
184 * :
185 * :
186 * :
187 * :
188 * :
189 * :
190 * :
191 * :
192 * :
193 * :
194 * :
195 * :
196 * :
197 * :
198 * :
199 * :
200 * :
201 * :
202 * :
203 * :
204 * :
205 * :
206 * :
207 * :
208 * :
209 * :
210 * :
211 * :
212 * :
213 * :
214 * :
215 * :
216 * :
217 * :
218 * :
219 * :
220 * :
221 * :
222 * :
223 * :
224 * :
225 * :
226 * :
227 * :
228 * :
229 * :
230 * :
231 * :
232 * :
233 * :
234 * :
235 * :
236 * :
237 * :
238 * :
239 * :
240 * :
241 * :
242 * :
243 * :
244 * :
245 * :
246 * :
247 * :
248 * :
249 * :
250 * :
251 * :
252 * :
253 * :
254 * :
255 * :
256 * :
257 * :
258 * :
259 * :
260 * :
261 * :
262 * :
263 * :
264 * :
265 * :
266 * :
267 * :
268 * :
269 * :
270 * :
271 * :
272 * :

```

```

195 * :
196 * :
197 * :
198 * :
199 * :
200 * :
201 * :
202 * :
203 * :
204 * :
205 * :
206 * :
207 * :
208 * :
209 * :
210 * :
211 * :
212 * :
213 * :
214 * :
215 * :
216 * :
217 * :
218 * :
219 * :
220 * :
221 * :
222 * :
223 * :
224 * :
225 * :
226 * :
227 * :
228 * :
229 * :
230 * :
231 * :
232 * :
233 * :
234 * :
235 * :
236 * :
237 * :
238 * :
239 * :
240 * :
241 * :
242 * :
243 * :
244 * :
245 * :
246 * :
247 * :
248 * :
249 * :
250 * :
251 * :
252 * :
253 * :
254 * :
255 * :
256 * :
257 * :
258 * :
259 * :
260 * :
261 * :
262 * :
263 * :
264 * :
265 * :
266 * :
267 * :
268 * :
269 * :
270 * :
271 * :
272 * :

```

```

273 *|*****
281 FUNCTION userinit
282 PARAMETERS notime
283 PUBLIC msafety, mexact, mescp, mescp, mproc, mpath
284 PUBLIC mudfparms, mbell, mscore, mdelete
285
286 mdelete = SET("DELETE")
287 msafety = SET("SAFETY")
288
289 => 2
290 mexact = SET("EXACT")
291
292 => 2
293 mescp = SET("ESCAPE")
294
295 => 2
296 mpath = SET("PATH")
297
298 => 2
299 mudfparms = SET("UDFPARMS")
300
301 => 2
302 mscore = SET("SCOREBOARD")
303
304 => 2
305 mbell = SET("BELL OFF")
306
307
308 SET RESOURCE TO camduser.dbf
309 SET SAFETY OFF
310 SET UDFPARMS TO REFERENCE
311 SET SCOREBOARD OFF
312 SET BELL OFF
313 SET ESCAPE ON
314
315 => pe key
316 SET DELETE ON
317
318 *|*****
319 * global user interface variables
320 *|*****
321 PUBLIC mtile, mversion, help_code, help_id, help_level, help_stack
322 DIMENSION help_stack(30)
323 PUBLIC singlebox
324
325 *|*****
326 * hypertext identifiers for on-line help
327 PUBLIC hyper_id
328 DIMENSION hyper_id(30)
329
330 => focus
331
332 mtile = "Computer Assisted Medical Diagnostic System"
333 mversion = "Version Release 1.0"
334 help_code = 0
335 help_level = 0
336
337 =inithelp()
338 =pushhelp("CAND")
339
340 hyper_id = ""
341 singlebox = CHR(218) + CHR(196) + CHR(191) + CHR(179) + ;
342           CHR(217) + CHR(196) + CHR(192) + CHR(179)
343
344 * default screen color set
345 PUBLIC c_deflt,c_disease,c_txt,c_err,c_area,c_pat
346 PUBLIC c_main,c_rev,c_treat,c_popup,c_rule,c_note
347 PUBLIC c_info,c_diag,c_help,c_prompt,c_getrev
348 PUBLIC c_encount,c_logon
349
350 * load optional color set
351 PRIVATE colorset
352 colorset = GETENV("COLORSET")
353 IF IEMPTY(colorset)
354   =getcolors(colorset)
355
356 *|*****
357 ENDIF

```

```

317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```



```

516 *| : POPHELP (procedure in U.PRG)
517 *|
518 *|*****
519 FUNCTION ednum *****
520 PARAMETERS mid, mname, mquestion, mextra, mhi, mlo, mwidth, mdec, mun
521 => its, mstrval
522 * mid dictionary id of item
523 * mname name of item
524 * mquestion user question text
525 * mextra additional user prompt text
526 * mhi hi limit advisory for value
527 * mlo lo limit advisory
528 * mwidth field width
529 * mdec number of decimals
530 * munits engineering units text
531 * mstrval current value
532
533 * save help context
534 PRIVATE oldhyp(alen(hyper_id))
535 =ACOPY(hyper_id, oldhyp)
536 hyper_id = ""
537 =pushhelp( mname )
538 hyper_id(1) = mname
539
540 PRIVATE mprompt
541 mprompt = mquestion
542 PRIVATE mevent
543 PRIVATE mbuttons
544 mevent = e_ok
545 mbuttons = 0
546 PRIVATE mval
547 mval = VAL(mstrval)
548
549 DO W6.SPR
550
551 DO CASE
552 CASE mbuttons = 1
553 mevent = e_ok
554 CASE mbuttons = 2
555 mevent = e_esc
556 ENDCASE
557
558 mstrval = LTRIM( STR( mval, mwidth, mdec ) )
559 =pophelp()
560 =ACOPY(oldhyp, hyper_id)
561 RETURN mevent
562
563 *****
564 *
565 * edit Boolean
566 *
567 *****
568 FUNCTION: ED800L()
569
570 Called by: EDITSYMP (procedure in DEMO.PRG)
571
572 Calls: PUSHHELP (procedure in U.PRG)
573 : W6C.SPR (procedure in U.PRG)
574 : POPHELP (procedure in U.PRG)
575
576 *****
577 FUNCTION edbool *****
578 PARAMETERS id, mname, mquestion, mextra, mstrval
579 * id dictionary id of item
580 * mname name of item
581 * mquestion user question text
582 * mextra additional user prompt text
583
584 *****

```

```

570 * mstrval current concatenated string value
571
572 * save help context
573 PRIVATE oldhyp(alen(hyper_id))
574 =ACOPY(hyper_id, oldhyp)
575 hyper_id = ""
576 =pushhelp( mname )
577 hyper_id(1) = mname
578
579 PRIVATE mevent
580 mprompt = mquestion
581 PRIVATE mchoice
582 PRIVATE mbuttons
583 mchoice = 1
584 mchoice = 1 IF (mstrval = "Y", 1, 2)
585 mbuttons = 0
586 mevent = e_ok
587
588 DO W6C.SPR
589
590 DO CASE
591 CASE mbuttons = 1
592 mevent = e_ok
593 CASE mbuttons = 2
594 mevent = e_esc
595 ENDCASE
596
597 IF mevent = e_esc
598 mstrval = "" IF (mchoice = 1, "YES", "NO")
599 ENDIF
600
601 =pophelp()
602 =ACOPY(oldhyp, hyper_id)
603 RETURN mevent
604
605 *****
606 *
607 * edit single choice checkbox
608 *
609 *****
610 Procedure: EDLIST *****
611
612 Calls: PUSHHELP (procedure in U.PRG)
613 : W6C.SPR (procedure in U.PRG)
614 : POPHELP (procedure in U.PRG)
615
616 *****
617 FUNCTION edlist *****
618 PARAMETERS id, mname, mquestion, mextra, choices, mstrval
619 * id dictionary id of item
620 * mname name of item
621 * mquestion user question text
622 * mextra additional user prompt text
623 * choices enumerated choices
624 * mstrval current concatenated string value
625
626 *****
627 EXTERNAL ARRAY choices *****
628
629 * save help context
630 PRIVATE oldhyp(alen(hyper_id))
631 =ACOPY(hyper_id, oldhyp)
632 hyper_id = ""
633 =pushhelp( mname )
634 hyper_id(1) = mname
635
636 * add hypertext entries to focus

```



```

636 PRIVATE i
637 FOR i = 1 TO ALEN(choices)
638   hyper_id[i + 1] = LTRIM(choices[i])
639   hyper_id[i] = ""
640   hyper_id[i] = ""
641   hyper_id[i] = ""
642   hyper_id[i] = ""
643   hyper_id[i] = ""
644   hyper_id[i] = ""
645   hyper_id[i] = ""
646   hyper_id[i] = ""
647   hyper_id[i] = ""
648   hyper_id[i] = ""
649   hyper_id[i] = ""
650   hyper_id[i] = ""
651   hyper_id[i] = ""
652   hyper_id[i] = ""
653   hyper_id[i] = ""
654   hyper_id[i] = ""
655   hyper_id[i] = ""
656   hyper_id[i] = ""
657   hyper_id[i] = ""
658   hyper_id[i] = ""
659   hyper_id[i] = ""
660   hyper_id[i] = ""
661   hyper_id[i] = ""
662   hyper_id[i] = ""
663   hyper_id[i] = ""
664   hyper_id[i] = ""
665   hyper_id[i] = ""
666   hyper_id[i] = ""
667   hyper_id[i] = ""
668   hyper_id[i] = ""
669   hyper_id[i] = ""
670   hyper_id[i] = ""
671   hyper_id[i] = ""
672   hyper_id[i] = ""
673   hyper_id[i] = ""
674   hyper_id[i] = ""
675   hyper_id[i] = ""
676   hyper_id[i] = ""
677   hyper_id[i] = ""
678   hyper_id[i] = ""
679   hyper_id[i] = ""
680   hyper_id[i] = ""
681   hyper_id[i] = ""
682   hyper_id[i] = ""
683   hyper_id[i] = ""
684   hyper_id[i] = ""
685   hyper_id[i] = ""
686   hyper_id[i] = ""
687   hyper_id[i] = ""
688   hyper_id[i] = ""
689   hyper_id[i] = ""
690   hyper_id[i] = ""
691   hyper_id[i] = ""
692   hyper_id[i] = ""
693   hyper_id[i] = ""
694   hyper_id[i] = ""
695   hyper_id[i] = ""
696   hyper_id[i] = ""
697   hyper_id[i] = ""
698   hyper_id[i] = ""
699   hyper_id[i] = ""

```

```

700 * save help context
701 PRIVATE oldhyp(alen(hyper_id))
702 =ACOPY(hyper_id, oldhyp)
703 hyper_id = ""
704 hyper_id = ""
705 hyper_id = ""
706 hyper_id = ""
707 hyper_id = ""
708 hyper_id = ""
709 hyper_id = ""
710 hyper_id = ""
711 hyper_id = ""
712 hyper_id = ""
713 hyper_id = ""
714 hyper_id = ""
715 hyper_id = ""
716 hyper_id = ""
717 hyper_id = ""
718 hyper_id = ""
719 hyper_id = ""
720 hyper_id = ""
721 hyper_id = ""
722 hyper_id = ""
723 hyper_id = ""
724 hyper_id = ""
725 hyper_id = ""
726 hyper_id = ""
727 hyper_id = ""
728 hyper_id = ""
729 hyper_id = ""
730 hyper_id = ""
731 hyper_id = ""
732 hyper_id = ""
733 hyper_id = ""
734 hyper_id = ""
735 hyper_id = ""
736 hyper_id = ""
737 hyper_id = ""
738 hyper_id = ""
739 hyper_id = ""
740 hyper_id = ""
741 hyper_id = ""
742 hyper_id = ""
743 hyper_id = ""
744 hyper_id = ""
745 hyper_id = ""
746 hyper_id = ""
747 hyper_id = ""
748 hyper_id = ""
749 hyper_id = ""
750 hyper_id = ""
751 hyper_id = ""
752 hyper_id = ""
753 hyper_id = ""
754 hyper_id = ""
755 hyper_id = ""
756 hyper_id = ""
757 hyper_id = ""
758 hyper_id = ""
759 hyper_id = ""
760 hyper_id = ""
761 hyper_id = ""
762 hyper_id = ""
763 hyper_id = ""
764 hyper_id = ""
765 hyper_id = ""
766 hyper_id = ""
767 hyper_id = ""
768 hyper_id = ""
769 hyper_id = ""
770 hyper_id = ""

```

END


```

943 Called by: W12.SPR
944
945 Calls: PUSHHELP
946 : U2.SPR
947 : POPHELP
948
949 *****
950 *****
951 *****
952 *****
953 *****
954 *****
955 *****
956 *****
957 *****
958 *****
959 *****
960 *****
961 *****
962 *****
963 *****
964 *****
965 *****
966 *****
967 *****
968 *****
969 *****
970 *****
971 *****
972 *****
973 *****
974 *****
975 *****
976 *****
977 *****
978 *****
979 *****
980 *****
981 *****
982 *****
983 *****
984 *****
985 *****
986 *****
987 *****
988 *****
989 *****
990 *****
991 *****
992 *****
993 *****
994 *****
995 *****
996 *****
997 *****
998 *****
999 *****
1000 *****
1001 *****
1002 *****
1003 *****
1004 *****

```

U.ACT 10-14-92 10:51a

Page 9 of 47


```

1483 *****
1474 FUNCTION myhandler
1475 PRIVATE m.temp
1476 IF dropped
1477 RETURN .T.
1478
1479 IF LEN(tobedone) > 0
1480 m.temp = tobedone
1481 tobedone = ""
1482 DO (m.temp) IN demo.prg
1483 RETURN .F.
1484
1485 IF
1486 CASE MONITOR("w encount")
1487 DO m.resume IN demo.prg
1488
1489 CASE MONITOR("w patedit")
1490 DO m.respat IN demo.prg
1491
1492 ENDIF
1493 RETURN .F.
1494 *****
1495
1496 Procedure: _MHIT
1497 Called by: CAND.MPR
1498 : GPOPLRB
1499 : CANDTR.MPR
1500 *****
1501
1502 PROCEDURE MHIT
1503 PARAMETERS PROG
1504 IF ROLEVEL() > 1
1505 tobedone = PROG
1506 CLEAR READ
1507 ELSE
1508 DO (PROG) IN demo.prg
1509
1510 hll01
1511 ENDIF
1512 *****
1513
1514 Procedure: _QUIT
1515 Called by: CAND.MPR
1516 : CANDTR.MPR
1517 *****
1518
1519 Calls: POPMESSAGE() (function in U.PRG)
1520 *****
1521
1522 PROCEDURE _QUIT
1523 IF EMPTY( WOUTPUT() )
1524
1525 dropped = .T.
1526 CLEAR READ ALL
1527 ELSE
1528
1529 ing"
1530
1531 popmessage("Close windows before quit")
1532
1533 ENDIF
1534 RETURN
1535 *****

```

U.ACT 10-14-92 10:51a

```

1532 *****
1533 * edit patient info
1534 *
1535 *****
1536
1537 Function: PATID()
1538 Called by: M_REGPAT (procedure in DEMO.PRG)
1539
1540 Calls: PUSHHELP (procedure in U.PRG)
1541 : W4.SPR
1542 : POPHELP (procedure in U.PRG)
1543 *****
1544
1545 FUNCTION PATID
1546 PARAMETERS mpatientid, mresocsec, mlastname, mfirst
1547
1548 * open a dialog window
1549 PRIVATE mbutton, mresult, mevent
1550 PRIVATE mbutton, mbutton
1551 mbutton = 0
1552 mbutton = 0
1553 PRIVATE quitting, saving
1554 quitting = .F.
1555 saving = .F.
1556 mevent = e_ok
1557 mbutton = e_ok
1558 mbutton = e_ok
1559 mbutton = e_ok
1560 mbutton = e_ok
1561
1562 DO W4.SPR
1563
1564 hll01
1565
1566 POPHELP()
1567
1568 IF mevent = e_ok .AND. saving
1569 mbutton = e_save
1570
1571 ENDIF
1572 RETURN mevent
1573 *****
1574
1575 * Push button controls
1576 *****
1577
1578 Procedure: _PVALID
1579 Called by: W4.SPR
1580 *****
1581
1582 FUNCTION _PVALID
1583
1584 DO CASE
1585 CASE mbutton = 1
1586 * OK button
1587 quitting = .T.
1588 IF mpatientid != 0
1589 mevent = e_ok
1590 ELSE
1591 * nothing selected
1592 mevent = e_esc
1593 ENDIF
1594 CLEAR READ
1595 CASE mbutton = 2
1596 * new button
1597 mevent = e_init
1598 quitting = .T.
1599
1600 ENDIF
1601 RETURN mevent
1602 *****

```

Page 13 of 47


```

1593 CLEAR READ
1594 CASE abuttons = 3
1595   * cancel button
1596   quitting = .T.
1597   mevent = e_esc
1598   CLEAR READ
1599   ENDCASE
1600   SHOW GETS
1601   RETURN
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751

```

CLEAR READ
 CASE abuttons = 3
 * cancel button
 quitting = .T.
 mevent = e_esc
 CLEAR READ
 ENDCASE
 SHOW GETS
 RETURN

* undo data popup menu screen

Function: UNDOMENU()
 Called by: MYUNDO
 Calls: W3.SPR

FUNCTION UNDOMENU
 * display menu prompts inside window
 PRIVATE mchoice, mevent, abuttons, apad
 DIMENSION mpad(4)
 mpad(1) = "Previous question"
 mpad(2) = "Continue current"
 mpad(3) = "Browse"
 mpad(4) = "Quit session"
 * get user's selection from menu
 mchoice = 0
 abuttons = 0
 mevent = e_ok
 DO W3.SPR
 ** h(10)
 ** go back to
 ** continue w
 ** review and
 ** quit sess
 ** default is
 ENDCASE
 RETURN mevent

Procedure: PUSHHELP
 Called by: WP2.PRG
 : USERINIT() (function in U.PRG)

```

1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751

```

: WPAT
 : EDITTEXT
 : BACKUP
 : RESTORE
 : ENCTITLE()
 : MCONFIRM()
 : PATID()
 : POPMESSAGE()
 : GETDISLIST()
 : GETAREALIST()
 : GETPATLIST()
 : GETENCLIST()
 : EDWML()
 : EDWBOX()
 : EDCHECK()
 : MCHOICE
 : GETDISLOCKUPLIST
 : EDLIST
 : _EVALID
 : _EDITVIOAG
 : DPGETPATLIST
 : SEARCH
 : STENCOUNT
 : STENCLIST
 : STGETPATLIST
 : _OPPOFLOT
 : SETHELP
 : (procedure in U.PRG)

FUNCTION PUSHHELP
 PARAMETERS helptopic
 IF help_level < ALEN(help_stack)
 help_level = help_level + 1
 ENDIF
 help_stack(help_level) = helptopic
 : sethelp()
 RETURN help_level

Procedure: POPHELP
 Called by: WP2.PRG
 : WPAT
 : EDITTEXT
 : BACKUP
 : RESTORE
 : ENCTITLE()
 : MCONFIRM()
 : PATID()
 : POPMESSAGE()
 : GETDISLIST()
 : GETAREALIST()
 : GETPATLIST()
 : GETENCLIST()
 : EDWML()
 : EDWBOX()
 : EDCHECK()
 : MCHOICE
 : GETDISLOCKUPLIST
 : EDLIST
 : _EVALID
 : _EDITVIOAG
 : DPGETPATLIST
 : SEARCH

U. ACT 10-14-92 10:51a

June 13 of 67


```

1933  => extract rulebase name and      mrulebase = TRIM(rulebase)      22
1934  => possible diagnosis threshold  cthreshold = threshold      22
1935  => probable diagnosis threshold  cthreshpro = probable      22
1936  => most likely diagnosis threshold  cthreshlik = likely      22
1937  => number of rules in rulebase  nrules = rules      22
1938  => close file                    _ENDIF
1939  => reselect original database      USE
1940  =>                                SELECT (msel)
1941  =>                                RETURN mrulebase
1942  =>                                _ENDIF
1943  =>                                RETURN mrulebase
1944  =>                                *
1945  =>                                * access colors database
1946  =>                                *
1947  =>                                *
1948  =>                                *
1949  =>                                *
1950  =>                                *
1951  =>                                * load optional color set
1952  =>                                *
1953  =>                                *
1954  =>                                *
1955  =>                                *
1956  =>                                *
1957  =>                                *
1958  =>                                *
1959  =>                                *
1960  =>                                *
1961  =>                                *
1962  =>                                *
1963  =>                                *
1964  =>                                *
1965  =>                                *
1966  =>                                *
1967  =>                                *
1968  =>                                *
1969  =>                                *
1970  =>                                *
1971  =>                                *
1972  =>                                *
1973  =>                                *
1974  =>                                *
1975  =>                                *
1976  =>                                *
1977  =>                                *
1978  =>                                *
1979  =>                                *
1980  =>                                *
1981  =>                                *
1982  =>                                *
1983  =>                                *
1984  =>                                *
1985  =>                                *
1986  =>                                *

```

```

1987  =>                                *
1988  =>                                *
1989  =>                                *
1990  =>                                *
1991  =>                                *
1992  =>                                *
1993  =>                                *
1994  =>                                *
1995  =>                                *
1996  =>                                *
1997  =>                                *
1998  =>                                *
1999  =>                                *
2000  =>                                *
2001  =>                                *
2002  =>                                *
2003  =>                                *
2004  =>                                *
2005  =>                                *
2006  =>                                *
2007  =>                                *
2008  =>                                *
2009  =>                                *
2010  =>                                *
2011  =>                                *
2012  =>                                *
2013  =>                                *
2014  =>                                *
2015  =>                                *
2016  =>                                *
2017  =>                                *
2018  =>                                *
2019  =>                                *
2020  =>                                *
2021  =>                                *
2022  =>                                *
2023  =>                                *
2024  =>                                *
2025  =>                                *
2026  =>                                *
2027  =>                                *
2028  =>                                *
2029  =>                                *
2030  =>                                *
2031  =>                                *
2032  =>                                *
2033  =>                                *
2034  =>                                *
2035  =>                                *
2036  =>                                *
2037  =>                                *
2038  =>                                *
2039  =>                                *
2040  =>                                *

```

```

2041 *|
2042 >> EOF()
2043 *|
2044 *|
2045 *|
2046 *|
2047 *|
2048 *|
2049 *|
2050 *|
2051 *|
2052 *|
2053 *|
2054 *|
2055 *|
2056 *|
2057 *|
2058 *|
2059 *|
2060 *|
2061 *|
2062 *|
2063 *|
2064 *|
2065 *|
2066 *|
2067 *|
2068 *|
2069 *|
2070 *|
2071 *|
2072 *|
2073 *|
2074 *|
2075 *|
2076 *|
2077 *|
2078 *|
2079 *|
2080 *|
2081 *|
2082 *|
2083 *|
2084 *|
2085 *|
2086 *|
2087 *|
2088 *|
2089 *|
2090 *|
2091 *|
2092 *|
2093 *|
2094 *|
2095 *|
2096 *|
2097 *|
2098 *|
2099 *|
2100 *|

```

Function: GETDIAGNOS()

Called by: LOENC (procedure in DEMO.PRG)

Calls: OPENDIAG (procedure in U.PRG)

: ADDDIAGNOS (procedure in DEMO.PRG)

FUNCTION getdiagnos

PARAMETERS mcountid

PRIVATE msel

msel = SELECT()

SCAN FOR id = mcountid

adddiagnos(STR(disease), probabi

ENDSCAN

USE

SELECT (msel)

RETURN .T.

* open diagnosis encounter file

* close data dictionary files

* close data dictionary files

Procedure: CLOSEDICT

Called by: FLUSHFILES() (function in U.PRG)

```

2101 *|
2102 *|
2103 *|
2104 *|
2105 *|
2106 *|
2107 *|
2108 *|
2109 *|
2110 *|
2111 *|
2112 *|
2113 *|
2114 *|
2115 *|
2116 *|
2117 *|
2118 *|
2119 *|
2120 *|
2121 *|
2122 *|
2123 *|
2124 *|
2125 *|
2126 *|
2127 *|
2128 *|
2129 *|
2130 *|
2131 *|
2132 *|
2133 *|
2134 *|
2135 *|
2136 *|
2137 *|
2138 *|
2139 *|
2140 *|
2141 *|
2142 *|
2143 *|
2144 *|
2145 *|
2146 *|
2147 *|
2148 *|
2149 *|
2150 *|
2151 *|
2152 *|
2153 *|
2154 *|
2155 *|
2156 *|
2157 *|

```

Procedure: OPENDIAG

Called by: PUTDIAGNOS() (function in U.PRG)

: GETDIAGNOS() (function in U.PRG)

FUNCTION opendia

SELECT 0

USE (f_diagnos) AGAIN INDEX (f_diagnos

RETURN SELECT()

* open encounter/diagnosis

* data base functions for data dictio

* open data dictionary files

* open data dictionary files

Procedure: OPENDICT

Called by: OPENFILES() (function in U.PRG)

Uses: DICT.DBF

: VAL.DBF

: ENUM.DBF

Aliases: DICTVAL

Indexes: DICTID.IDX

: DICTNAME.IDX

: VAL.IDX

: ENUM.IDX

FUNCTION opendict

SELECT 0

USE dict AGAIN INDEX dictid,dictname

SELECT 0

USE VAL AGAIN INDEX VAL ALIAS dictval

SELECT 0

USE enum AGAIN INDEX enum

RETURN .T.

* close data dictionary files

* close data dictionary files

Procedure: CLOSEDICT

Called by: FLUSHFILES() (function in U.PRG)

```

2158 *| *****
2159 | FUNCTION closedict
2160 | PRIVATE msel
2161 | msel = SELECT()
2162 |
2163 | SELECT dict
2164 | USE
2165 | SELECT dictval
2166 | USE
2167 | SELECT enum
2168 | USE
2169 | SELECT (msel)
2170 | RETURN .T.
2171 |
2172 | *****
2173 | * size data dictionary
2174 | *
2175 | *****
2176 |
2177 | Function: DICTCOUNT()
2178 | Called by: INITSYMP() (function in DEMO.PRG)
2179 | Uses: DICT.DBF
2180 | *****
2181 | FUNCTION dictcount
2182 | PRIVATE msel
2183 | msel = SELECT()
2184 |
2185 | SELECT 0
2186 | USE dict AGAIN
2187 |
2188 | *****
2189 | * save current database selection
2190 | *
2191 | *****
2192 | * list of elements
2193 | *
2194 | *****
2195 | * number of names in file
2196 | *
2197 | *****
2198 | * close file
2199 | *
2200 | *****
2201 | * reselct original database
2202 | *
2203 | *****
2204 | RETURN nconditions
2205 |
2206 | *****
2207 | * Lookup Dictionary item
2208 | *
2209 | *****
2210 | Function: DICTQV()
2211 | Called by: LOOKUPQ
2212 | (procedure in DEMO.PRG)
2213 | : LOOKUPNAM
2214 | : EDITSYMP
2215 | : PUTSYMP()
2216 | : GETSYMP()

```

```

2217 *| *****
2218 | FUNCTION dictq
2219 | PARAMETERS mid, mask, mq, aname, mpromp
2220 |
2221 | PRIVATE mtype
2222 | PRIVATE msel
2223 | msel = SELECT()
2224 |
2225 | *****
2226 | * current database selection
2227 | *
2228 | *****
2229 | * look up identifier in dictionary
2230 | *
2231 | *****
2232 | * return associated attributes
2233 | *
2234 | *****
2235 | * mtype = datatype
2236 | *
2237 | *****
2238 | * IF PARAMETERS() > 1
2239 | * mask = askable
2240 | *
2241 | *****
2242 | * IF PARAMETERS() > 2
2243 | * mq = TRIM(question)
2244 | *
2245 | *****
2246 | * IF PARAMETERS() > 3
2247 | * aname = name
2248 | *
2249 | *****
2250 | * IF PARAMETERS() > 4
2251 | * mpromp = TRIM(PROMPT)
2252 | *
2253 | *****
2254 | * ELSE
2255 | * mtype = ""
2256 | *
2257 | *****
2258 | * ENDIF
2259 | *
2260 | *****
2261 | * SELECT (msel)
2262 | *
2263 | *****
2264 | * RETURN mtype
2265 | *
2266 | *****
2267 | * Lookup Dictionary numeric value item
2268 | *
2269 | *****
2270 |
2271 | *****
2272 | Function: DICTQV()
2273 | Called by: EDITSYMP
2274 | (procedure in DEMO.PRG)
2275 | *****
2276 | FUNCTION dictq
2277 | PARAMETERS mid, mhi, mlo, mwidth, mdec,
2278 |
2279 | PRIVATE mresult
2280 | PRIVATE msel
2281 | msel = SELECT()
2282 |
2283 | *****
2284 | * current database selection
2285 | *
2286 | *****
2287 | * look up identifier in dictionary
2288 | *
2289 | *****
2290 | * IF FOUND()
2291 | * mresult = .T.
2292 | *
2293 | *****
2294 | * mhi = hi
2295 | *
2296 | *****
2297 | * mlo = lo
2298 | *
2299 | *****
2300 | * mwidth = WIDTH
2301 | *
2302 | *****
2303 | * mdec = dec
2304 | *
2305 | *****

```

```

2265      units = TRIM(units)
2266      ELSE mresult = .F.
2267      ENDIF
2268      SELECT (msel)
2269  => restore database selection
2270      RETURN mresult
2271
2272  =>
2273
2274  * size dictionary enumerated item
2275  *
2276  *****
2277  Function: DICTCOUNT()
2278  Called by: EDITSYMP (procedure in DEMO.PRG)
2279  *****
2280  FUNCTION DICTCOUNT
2281  PARAMETERS mid
2282  PRIVATE N
2283  PRIVATE mkey
2284  PRIVATE msel
2285  msel = SELECT()
2286
2287  => save current database selection
2288  SELECT enum
2289  * look up identifier in dictionary
2290  N = 0
2291  mkey = VAL(mid)
2292  SEEK mkey
2293  * count list of names
2294  COUNT WHILE mkey = id .AND. .NOT. EOF()
2295  TO N
2296
2297  =>
2298  SELECT (msel)
2299  RETURN N
2300
2301  =>
2302  * Lookup Dictionary enumerated value it
2303  *
2304  *
2305  *****
2306  Function: DICTQ()
2307  Called by: EDITSYMP (procedure in DEMO.PRG)
2308  *****
2309  FUNCTION DICTQ
2310  PARAMETERS mid, names, attrib
2311  EXTERNAL ARRAY names, attrib
2312  PRIVATE I
2313  PRIVATE mkey
2314  PRIVATE msel
2315
2316  =>
2317
2318  =>
2319
2320

```

```

2321  => current database selection
2322  msel = SELECT()
2323  SELECT enum
2324  * look up identifier in dictionary
2325  i = 0
2326  mkey = VAL(mid)
2327  SEEK mkey
2328  IF FOUND()
2329  * create list of names
2330  i = 1
2331  DO WHILE mkey = id .AND. .NOT. EOF()
2332  names(i) = TRIM(enumerate)
2333  attrib(i) = mkey
2334  i = i + 1
2335  SKIP
2336  ENDIF
2337  SELECT (msel)
2338  RETURN 1
2339
2340  => restore database selection
2341
2342  *
2343  *
2344  * Lookup Dictionary enumerated item ord
2345  *
2346  *****
2347  Procedure: DICTORD
2348  Called by: PUTSYMPOM() (function in U.PRG)
2349  *****
2350  FUNCTION DICTORD
2351  PARAMETERS mid, msel
2352  PRIVATE mkey, word
2353  PRIVATE msel
2354  msel = SELECT()
2355
2356  => current database selection
2357  SELECT enum
2358  * look up identifier in dictionary
2359  mkey = VAL(mid)
2360  word = 0
2361  SEEK mkey
2362  IF FOUND()
2363  * find enumeration
2364  DO WHILE mkey = id .AND. .NOT. EOF()
2365  word = word + 1
2366  IF word = ord
2367  RETURN word
2368
2369  => corresponding ordinal
2370  EXIT
2371  ENDIF
2372  SKIP
2373  ENDIF
2374  SELECT (msel)
2375  RETURN ALLTRIM(STR(word))
2376
2377  => restore database selection

```



```

2496 *|
2497 *|      Called by: ADDHISTORY      (procedure in DEMO.PRG)
2498 *|
2499 *|*****
2500 *|      FUNCTION dictionaryname
2501 *|      PARAMETERS wname
2502 *|      PRIVATE mid
2503 *|      PRIVATE msel
2504 *|      msel = SELECT()
2505 *|
2506 *|      SELECT dict
2507 *|      SET ORDER TO 2
2508 *|
2509 *|      SEEK wname
2510 *|      IF FOUND()
2511 *|      mid = ALLTRIM(STR(id))
2512 *|      ELSE
2513 *|      mid = ""
2514 *|      ENDIF
2515 *|      SET ORDER TO 1
2516 *|      SELECT (msel)
2517 *|
2518 *|      RETURN mid
2519 *|
2520 *|*****
2521 *|
2522 *|      * data base functions for patient encou
2523 *|
2524 *|      *
2525 *|
2526 *|      *
2527 *|
2528 *|      *
2529 *|
2530 *|      * open encounter files
2531 *|      *
2532 *|      *
2533 *|
2534 *|*****
2535 *|
2536 *|      Procedure: OPENENCNT
2537 *|
2538 *|      Called by: OPENFILES()      (function in U.PRG)
2539 *|
2540 *|*****
2541 *|      FUNCTION openencnt
2542 *|
2543 *|      PUBLIC fencntr
2544 *|      IF USED(f_encntr)
2545 *|      SELECT f
2546 *|      USE (f_encntr) AGAIN INDEX (f_enco
2547 *|      fencntr = SELECT()
2548 *|      SELECT (f_encntr)
2549 *|      fencntr = SELECT()
2550 *|      ENDIF
2551 *|      RETURN .T.
2552 *|
2553 *|*****
2554 *|      * flush encounter files
2555 *|
2556 *|
2557 *|
2558 *|
2559 *|
2560 *|
2561 *|
2562 *|
2563 *|
2564 *|
2565 *|
2566 *|
2567 *|
2568 *|
2569 *|
2570 *|
2571 *|
2572 *|
2573 *|
2574 *|
2575 *|
2576 *|
2577 *|
2578 *|
2579 *|
2580 *|
2581 *|
2582 *|
2583 *|
2584 *|
2585 *|
2586 *|
2587 *|
2588 *|
2589 *|
2590 *|
2591 *|
2592 *|
2593 *|
2594 *|
2595 *|
2596 *|
2597 *|
2598 *|
2599 *|
2600 *|
2601 *|

```

```

2545 *|
2546 *|
2547 *|
2548 *|
2549 *|
2550 *|
2551 *|
2552 *|
2553 *|
2554 *|
2555 *|
2556 *|
2557 *|
2558 *|
2559 *|
2560 *|
2561 *|
2562 *|
2563 *|
2564 *|
2565 *|
2566 *|
2567 *|
2568 *|
2569 *|
2570 *|
2571 *|
2572 *|
2573 *|
2574 *|
2575 *|
2576 *|
2577 *|
2578 *|
2579 *|
2580 *|
2581 *|
2582 *|
2583 *|
2584 *|
2585 *|
2586 *|
2587 *|
2588 *|
2589 *|
2590 *|
2591 *|
2592 *|
2593 *|
2594 *|
2595 *|
2596 *|
2597 *|
2598 *|
2599 *|
2600 *|
2601 *|

```

```

2607 * Procedure: ENCID
2608 *
2609 *
2610 *
2611 * FUNCTION ENCID
2612 * PARAMETERS mpatientid, mdate, mtime
2613 *
2614 * PRIVATE mid
2615 * mid = 0
2616 * PRIVATE msel
2617 * msel = SELECT()
2618 *
2619 * SELECT (fencounter)
2620 * SEEK mpatientid
2621 * IF FOUND()
2622 * DO WHILE .NOT. EOF() .AND. patient =
2623 * IF mdate = DATE
2624 * * found encounter by date opti
2625 * ONLY match time
2626 * IF TYPE("mtime")="C",mtime
2627 * = TIME.T.)
2628 * * found (first or exact) en
2629 * counter record
2630 * mid = encounterid
2631 * EXIT
2632 * ENDIF
2633 * SKIP
2634 * ENDIF
2635 * ENDIF
2636 * ENDIF
2637 * ENDIF
2638 * ENDIF
2639 * ENDIF
2640 * ENDIF
2641 * ENDIF
2642 * ENDIF
2643 * ENDIF
2644 * ENDIF
2645 * ENDIF
2646 * ENDIF
2647 * ENDIF
2648 * ENDIF
2649 * ENDIF
2650 * ENDIF
2651 * ENDIF
2652 * ENDIF
2653 * ENDIF
2654 * ENDIF
2655 * ENDIF
2656 * ENDIF
2657 * ENDIF
2658 * ENDIF
2659 * ENDIF
2660 * ENDIF
2661 * ENDIF
2662 * ENDIF
2663 * ENDIF
2664 * ENDIF
2665 * ENDIF
2666 * ENDIF
2667 * ENDIF
2668 * ENDIF
2669 * ENDIF
2670 * ENDIF
2671 * ENDIF
2672 * ENDIF
2673 * ENDIF
2674 * ENDIF
2675 * ENDIF
2676 * ENDIF
2677 * ENDIF
2678 * ENDIF
2679 * ENDIF
2680 * ENDIF
2681 * ENDIF
2682 * ENDIF
2683 * ENDIF
2684 * ENDIF
2685 * ENDIF
2686 * ENDIF
2687 * ENDIF
2688 * ENDIF
2689 * ENDIF
2690 * ENDIF
2691 * ENDIF
2692 * ENDIF
2693 * ENDIF
2694 * ENDIF
2695 * ENDIF
2696 * ENDIF
2697 * ENDIF
2698 * ENDIF
2699 * ENDIF
2700 * ENDIF
2701 * ENDIF
2702 * ENDIF
2703 * ENDIF
2704 * ENDIF
2705 * ENDIF
2706 * ENDIF
2707 * ENDIF
2708 * ENDIF
2709 * ENDIF
2710 * ENDIF
2711 * ENDIF
2712 * ENDIF
2713 * ENDIF
2714 * ENDIF
2715 * ENDIF
2716 * ENDIF
2717 * ENDIF
2718 * ENDIF
2719 * ENDIF
2720 * ENDIF
2721 * ENDIF
2722 * ENDIF
2723 * ENDIF
2724 * ENDIF
2725 * ENDIF
2726 * ENDIF
2727 * ENDIF
2728 * ENDIF
2729 * ENDIF
2730 * ENDIF
2731 * ENDIF
2732 * ENDIF
2733 * ENDIF
2734 * ENDIF
2735 * ENDIF
2736 * ENDIF
2737 * ENDIF
2738 * ENDIF
2739 * ENDIF
2740 * ENDIF
2741 * ENDIF
2742 * ENDIF
2743 * ENDIF
2744 * ENDIF
2745 * ENDIF
2746 * ENDIF
2747 * ENDIF
2748 * ENDIF
2749 * ENDIF
2750 * ENDIF
2751 * ENDIF
2752 * ENDIF
2753 * ENDIF
2754 * ENDIF
2755 * ENDIF
2756 * ENDIF
2757 * ENDIF
2758 * ENDIF
2759 * ENDIF
2760 * ENDIF
2761 * ENDIF
2762 * ENDIF
2763 * ENDIF
2764 * ENDIF
2765 * ENDIF
2766 * ENDIF
2767 * ENDIF
2768 * ENDIF
2769 * ENDIF
2770 * ENDIF
2771 * ENDIF
2772 * ENDIF
2773 * ENDIF
2774 * ENDIF
2775 * ENDIF
2776 * ENDIF
2777 * ENDIF
2778 * ENDIF
2779 * ENDIF
2780 * ENDIF
2781 * ENDIF
2782 * ENDIF
2783 * ENDIF
2784 * ENDIF
2785 * ENDIF
2786 * ENDIF
2787 * ENDIF
2788 * ENDIF
2789 * ENDIF
2790 * ENDIF
2791 * ENDIF
2792 * ENDIF
2793 * ENDIF
2794 * ENDIF
2795 * ENDIF
2796 * ENDIF
2797 * ENDIF
2798 * ENDIF
2799 * ENDIF
2800 * ENDIF
2801 * ENDIF
2802 * ENDIF
2803 * ENDIF
2804 * ENDIF
2805 * ENDIF
2806 * ENDIF
2807 * ENDIF
2808 * ENDIF
2809 * ENDIF
2810 * ENDIF
2811 * ENDIF
2812 * ENDIF
2813 * ENDIF
2814 * ENDIF
2815 * ENDIF
2816 * ENDIF
2817 * ENDIF
2818 * ENDIF
2819 * ENDIF
2820 * ENDIF
2821 * ENDIF
2822 * ENDIF
2823 * ENDIF
2824 * ENDIF
2825 * ENDIF
2826 * ENDIF
2827 * ENDIF
2828 * ENDIF
2829 * ENDIF
2830 * ENDIF
2831 * ENDIF
2832 * ENDIF
2833 * ENDIF
2834 * ENDIF
2835 * ENDIF
2836 * ENDIF
2837 * ENDIF
2838 * ENDIF
2839 * ENDIF
2840 * ENDIF
2841 * ENDIF
2842 * ENDIF
2843 * ENDIF
2844 * ENDIF
2845 * ENDIF
2846 * ENDIF
2847 * ENDIF
2848 * ENDIF
2849 * ENDIF
2850 * ENDIF
2851 * ENDIF
2852 * ENDIF
2853 * ENDIF
2854 * ENDIF
2855 * ENDIF
2856 * ENDIF
2857 * ENDIF
2858 * ENDIF
2859 * ENDIF
2860 * ENDIF
2861 * ENDIF
2862 * ENDIF
2863 * ENDIF
2864 * ENDIF
2865 * ENDIF
2866 * ENDIF
2867 * ENDIF
2868 * ENDIF
2869 * ENDIF
2870 * ENDIF
2871 * ENDIF
2872 * ENDIF
2873 * ENDIF
2874 * ENDIF
2875 * ENDIF
2876 * ENDIF
2877 * ENDIF
2878 * ENDIF
2879 * ENDIF
2880 * ENDIF
2881 * ENDIF
2882 * ENDIF
2883 * ENDIF
2884 * ENDIF
2885 * ENDIF
2886 * ENDIF
2887 * ENDIF
2888 * ENDIF
2889 * ENDIF
2890 * ENDIF
2891 * ENDIF
2892 * ENDIF
2893 * ENDIF
2894 * ENDIF
2895 * ENDIF
2896 * ENDIF
2897 * ENDIF
2898 * ENDIF
2899 * ENDIF
2900 * ENDIF
2901 * ENDIF
2902 * ENDIF
2903 * ENDIF
2904 * ENDIF
2905 * ENDIF
2906 * ENDIF
2907 * ENDIF
2908 * ENDIF
2909 * ENDIF
2910 * ENDIF
2911 * ENDIF
2912 * ENDIF
2913 * ENDIF
2914 * ENDIF
2915 * ENDIF
2916 * ENDIF
2917 * ENDIF
2918 * ENDIF
2919 * ENDIF
2920 * ENDIF
2921 * ENDIF
2922 * ENDIF
2923 * ENDIF
2924 * ENDIF
2925 * ENDIF
2926 * ENDIF
2927 * ENDIF
2928 * ENDIF
2929 * ENDIF
2930 * ENDIF
2931 * ENDIF
2932 * ENDIF
2933 * ENDIF
2934 * ENDIF
2935 * ENDIF
2936 * ENDIF
2937 * ENDIF
2938 * ENDIF
2939 * ENDIF
2940 * ENDIF
2941 * ENDIF
2942 * ENDIF
2943 * ENDIF
2944 * ENDIF
2945 * ENDIF
2946 * ENDIF
2947 * ENDIF
2948 * ENDIF
2949 * ENDIF
2950 * ENDIF
2951 * ENDIF
2952 * ENDIF
2953 * ENDIF
2954 * ENDIF
2955 * ENDIF
2956 * ENDIF
2957 * ENDIF
2958 * ENDIF
2959 * ENDIF
2960 * ENDIF
2961 * ENDIF
2962 * ENDIF
2963 * ENDIF
2964 * ENDIF
2965 * ENDIF
2966 * ENDIF
2967 * ENDIF
2968 * ENDIF
2969 * ENDIF
2970 * ENDIF
2971 * ENDIF
2972 * ENDIF
2973 * ENDIF
2974 * ENDIF
2975 * ENDIF
2976 * ENDIF
2977 * ENDIF
2978 * ENDIF
2979 * ENDIF
2980 * ENDIF
2981 * ENDIF
2982 * ENDIF
2983 * ENDIF
2984 * ENDIF
2985 * ENDIF
2986 * ENDIF
2987 * ENDIF
2988 * ENDIF
2989 * ENDIF
2990 * ENDIF
2991 * ENDIF
2992 * ENDIF
2993 * ENDIF
2994 * ENDIF
2995 * ENDIF
2996 * ENDIF
2997 * ENDIF
2998 * ENDIF
2999 * ENDIF
3000 * ENDIF

```

```

2656 * SELECT (fencounter)
2657 * SET ORDER TO 3
2658 *
2659 * * look up encounter record
2660 * SEEK mpatientid
2661 * IF FOUND()
2662 * mresult = .T.
2663 * REPLACE DATE WITH mdateenc
2664 * REPLACE TIME WITH mtimeenc
2665 * REPLACE station WITH mstation
2666 * REPLACE department WITH mdept
2667 * REPLACE provider WITH mprovider
2668 * REPLACE area WITH marea
2669 * REPLACE diagnosis WITH mdiagnosis
2670 * REPLACE treatment WITH mtreatment
2671 * REPLACE state WITH mstate
2672 * ENDIF
2673 * SET ORDER TO 1
2674 *
2675 * * restore
2676 *
2677 * * restore
2678 *
2679 * * Get encounter info
2680 *
2681 *
2682 *
2683 *
2684 *
2685 *
2686 *
2687 *
2688 *
2689 *
2690 *
2691 *
2692 *
2693 *
2694 *
2695 *
2696 *
2697 *
2698 *
2699 *
2700 *
2701 *
2702 *
2703 *
2704 *
2705 *
2706 *
2707 *
2708 *
2709 *
2710 *
2711 *
2712 *
2713 *
2714 *
2715 *
2716 *
2717 *
2718 *
2719 *
2720 *
2721 *
2722 *
2723 *
2724 *
2725 *
2726 *
2727 *
2728 *
2729 *
2730 *
2731 *
2732 *
2733 *
2734 *
2735 *
2736 *
2737 *
2738 *
2739 *
2740 *
2741 *
2742 *
2743 *
2744 *
2745 *
2746 *
2747 *
2748 *
2749 *
2750 *
2751 *
2752 *
2753 *
2754 *
2755 *
2756 *
2757 *
2758 *
2759 *
2760 *
2761 *
2762 *
2763 *
2764 *
2765 *
2766 *
2767 *
2768 *
2769 *
2770 *
2771 *
2772 *
2773 *
2774 *
2775 *
2776 *
2777 *
2778 *
2779 *
2780 *
2781 *
2782 *
2783 *
2784 *
2785 *
2786 *
2787 *
2788 *
2789 *
2790 *
2791 *
2792 *
2793 *
2794 *
2795 *
2796 *
2797 *
2798 *
2799 *
2800 *
2801 *
2802 *
2803 *
2804 *
2805 *
2806 *
2807 *
2808 *
2809 *
2810 *
2811 *
2812 *
2813 *
2814 *
2815 *
2816 *
2817 *
2818 *
2819 *
2820 *
2821 *
2822 *
2823 *
2824 *
2825 *
2826 *
2827 *
2828 *
2829 *
2830 *
2831 *
2832 *
2833 *
2834 *
2835 *
2836 *
2837 *
2838 *
2839 *
2840 *
2841 *
2842 *
2843 *
2844 *
2845 *
2846 *
2847 *
2848 *
2849 *
2850 *
2851 *
2852 *
2853 *
2854 *
2855 *
2856 *
2857 *
2858 *
2859 *
2860 *
2861 *
2862 *
2863 *
2864 *
2865 *
2866 *
2867 *
2868 *
2869 *
2870 *
2871 *
2872 *
2873 *
2874 *
2875 *
2876 *
2877 *
2878 *
2879 *
2880 *
2881 *
2882 *
2883 *
2884 *
2885 *
2886 *
2887 *
2888 *
2889 *
2890 *
2891 *
2892 *
2893 *
2894 *
2895 *
2896 *
2897 *
2898 *
2899 *
2900 *
2901 *
2902 *
2903 *
2904 *
2905 *
2906 *
2907 *
2908 *
2909 *
2910 *
2911 *
2912 *
2913 *
2914 *
2915 *
2916 *
2917 *
2918 *
2919 *
2920 *
2921 *
2922 *
2923 *
2924 *
2925 *
2926 *
2927 *
2928 *
2929 *
2930 *
2931 *
2932 *
2933 *
2934 *
2935 *
2936 *
2937 *
2938 *
2939 *
2940 *
2941 *
2942 *
2943 *
2944 *
2945 *
2946 *
2947 *
2948 *
2949 *
2950 *
2951 *
2952 *
2953 *
2954 *
2955 *
2956 *
2957 *
2958 *
2959 *
2960 *
2961 *
2962 *
2963 *
2964 *
2965 *
2966 *
2967 *
2968 *
2969 *
2970 *
2971 *
2972 *
2973 *
2974 *
2975 *
2976 *
2977 *
2978 *
2979 *
2980 *
2981 *
2982 *
2983 *
2984 *
2985 *
2986 *
2987 *
2988 *
2989 *
2990 *
2991 *
2992 *
2993 *
2994 *
2995 *
2996 *
2997 *
2998 *
2999 *
3000 *

```



```

2828      ENDIF
2829      => remove filter condition
2830
2831      => restore index order
2832
2833      => restore database selection
2834      RETURN 1
2835
2836      ** list of names
2837
2838      ** get disease descriptions
2839
2840      *****
2841      *****
2842      *****
2843      *****
2844      *****
2845      *****
2846      *****
2847      *****
2848      *****
2849      *****
2850      *****
2851      *****
2852      *****
2853      *****
2854      *****
2855      *****
2856      *****
2857      *****
2858      *****
2859      *****
2860      *****
2861      *****
2862      *****
2863      *****
2864      *****
2865      *****
2866      *****
2867      *****
2868      *****
2869      *****
2870      *****
2871      *****
2872      *****
2873      *****
2874      *****
2875      *****
2876      *****
2877      *****
2878      *****
2879      *****
2880      *****

```

```
*****  
***** | | | | | *****  
Function: DISEASECNT()  
  
Called by: INDIAGNOS()      (function in DEMO.PRG)  
          : ASKDIAG        (procedure in DEMO.PRG)  
          : M_LOOKUP       (procedure in DEMO.PRG)  
  
Uses: DISEASE.DBF  
  
*****  
FUNCTION diseasecnt  
PRIVATE N  
PRIVATE msel  
msel = SELECT()  
  
    ** save current database selection  
    ** list of elements  
    ** number of names in file  
    ** close file  
    ** resselect original database  
RETURN N  
  
*****  
***** * * * * *  
** get index of disease topics  
** * * * * *  
  
*****  
Function: DISNAMCS()  
  
Called by: ASKDIAI         (procedure in DEMO.PRG)  
          : M_LOOKUP       (procedure in DEMO.PRG)  
  
Uses: DISEASE.DBF  
  
Indexes: DISID.IDX  
          : DISEASE.IDX  
  
*****  
PROCEDURE disnames  
PARAMETERS mdiseases, marec  
EXTERNAL ARRAY mdiseases  
  
PRIVATE i  
PRIVATE msel  
msel = SELECT()  
  
    SELECT 0  
    USE disease  
IF PARAMETERS(i) = 2  
SET INDEX TO d'id  
| SET FILTER TO area = marec  
GOTO TOP  
  
*****  
***** save c  
***** diseas
```

```

2935      ELSE
2936      SET INDEX TO disease
2937      ENDIF
2938
2939      * create list of names
2940      i = 0
2941      DO WHILE .NOT. EOF()
2942      i = i + 1
2943      mdiseases[i] = TRIM( name )
2944      SKIP
2945      ENDDO
2946      DECLARE mdiseases[]
2947      USE
2948      SELECT (msel)
2949      => ** restore database selection
2950      RETURN 1
2951      => ** list of names
2952
2953      * update disease descriptions
2954      *
2955
2956      *****
2957      Procedure: DISEASEPUT
2958      Uses: DISEASE.DBF
2959      Indexes: DISEASE.IDX
2960
2961      *****
2962      FUNCTION diseaseput
2963      PARAMETERS mdisease, mhelp
2964      PRIVATE msel
2965      msel = SELECT()
2966      => e current database selection
2967
2968      SELECT 0
2969      USE disease INDEX disease
2970      SEEK UPPER(mdisease)
2971      IF FOUND()
2972      REPLACE DESCRPT WITH mhelp
2973      ENDIF
2974      USE
2975      SELECT (msel)
2976      => select original database
2977
2978      RETURN **
2979
2980      *
2981
2982      * get disease name
2983      *
2984      *
2985      *****
2986      Function: DISEASEMAN()
2987      Called by: BATCHEST()
2988      (function in DEMO.PRG)
2989      (procedure in DEMO.PRG)
2990      (procedure in DEMO.PRG)
2991
2992      *****

```

```

3008      * STRDIAG
3009      * _EDITDIAG
3010      *
3011      Uses: DISEASE.DBF
3012
3013      Indexes: DISID.IDX
3014
3015      *****
3016      FUNCTION diseaseman
3017      PARAMETERS mid, mname
3018      EXTERNAL ARRAY mid, mname
3019      => ** save current
3020
3021      PRIVATE i, mresult
3022      mresult = ""
3023      PRIVATE msel
3024      msel = SELECT()
3025
3026      SELECT 0
3027      USE disease INDEX disid
3028
3029      IF PARAMETERS() = 2
3030      * translate list
3031      FOR i = 1 TO ALEN(mid)
3032      SEEK mid[i]
3033      mname[i] = IIF( FOUND(), TRIM(
3034      name ) , "" )
3035      L-NEXT
3036      ELSE
3037      SEEK mid
3038      mresult = IIF( FOUND(), TRIM( n
3039      ame ) , "" )
3040      ENDIF
3041
3042      * reslect original database
3043      RETURN mresult
3044
3045      *
3046      * get disease id
3047      *
3048      *****
3049      Function: DISEASEID()
3050      Called by: M_DIAG
3051      (procedure in DEMO.PRG)
3052      Uses: DISEASE.DBF
3053      Indexes: DISEASE.IDX
3054
3055      *****
3056      FUNCTION diseasedid
3057      PARAMETERS dname, did
3058      EXTERNAL ARRAY did, dname
3059      => ** rwd007

```

```

3050 PRIVATE i, mresult
3051 mresult = ""
3052 PRIVATE msel
3053 msel = SELECT()
3054 => ent database selection
3055 SELECT 0
3056 USE disease INDEX disease
3057
3058 IF PARAMETERS() = 2
3059   FOR i = 1 TO ALEN(chname)
3060     SEEK chname[i]
3061     didit() = IIF(FOUND(), id, 0)
3062     L-NEXT
3063   ELSE
3064     SEEK chname
3065     mresult = IIF(FOUND(), id, 0)
3066   L-ENDIF
3067
3068 USE
3069 SELECT (msel)
3070
3071 RETURN mresult
3072
3073 ***** Expected NEXT--Found ENDF
3074
3075 ***** reselct original database
3076
3077 *****
3078 *****
3079 *****
3080 *****
3081 *****
3082 *****
3083 *****
3084 *****
3085 *****
3086 *****
3087 *****
3088 *****
3089 *****
3090 *****
3091 *****
3092 *****
3093 *****
3094 *****
3095 *****
3096 *****
3097 *****
3098 *****
3099 *****
3100 *****
3101 *****
3102 *****
3103 *****
3104 *****
3105 *****
3106 *****
3107 *****
3108 *****
3109 *****
3110 *****
3111 *****
3112 *****
3113 *****
3114 *****
3115 *****
3116 *****
3117 *****
3118 *****
3119 *****
3120 *****
3121 *****
3122 *****
3123 *****
3124 *****
3125 *****
3126 *****
3127 *****
3128 *****
3129 *****
3130 *****
3131 *****
3132 *****
3133 *****
3134 *****
3135 *****
3136 *****
3137 *****
3138 *****
3139 *****
3140 *****
3141 *****
3142 *****

```

```

3105 PUBLIC f_patname
3106 PUBLIC f_patid
3107 PUBLIC f_encountnr
3108 PUBLIC f_encountd
3109 PUBLIC f_encounti
3110 f_symptom = "symptoms"
3111 f_diagnose = "diagnose"
3112 f_patient = "patient"
3113 f_patname = "patname"
3114 f_patid = "patid"
3115 f_encountnr = "encountnr"
3116 f_encountd = "encountd"
3117 f_encounti = "encounti"
3118
3119 IF EMPTY(testing)
3120   * use test files
3121   IF RIGHT(testing, 1) != "\
3122     testing = testing + "\
3123   L-ENDIF
3124   f_symptom = testing + f_symp
3125   f_diagnose = testing + f_dia
3126   f_patient = testing + f_patie
3127   f_patname = testing + f_patna
3128   f_patid = testing + f_patid
3129   f_encountnr = testing + f_encou
3130   f_encountd = testing + f_encou
3131   f_encounti = testing + f_encou
3132 L-ENDIF
3133
3134 =openpat()
3135 =openencnt()
3136 =opendict()
3137 =openrule()
3138 IF EMPTY(testing)
3139   =popuphide(m.wpop)
3140 L-ENDIF
3141 RETURN .T.
3142

```



```

3375 *****
3376 Function: NEWPATIENT()
3377
3378 Called by: M_PATADO (procedure in DEMO.PRG)
3379
3380 *****
3381 FUNCTION newpatient
3382
3383   strname, mfirstname, mdatebirth, msex, mwaterank
3384   PARAMETERS mpatientid, msecsec, mla
3385
3386   PRIVATE msec
3387   msec = SELECT()
3388
3389   save current database selection
3390
3391   SELECT (fpatient)
3392   APPEND BLANK
3393
3394   REPLACE id WITH mpatientid
3395   REPLACE lastname WITH mlastname,
3396   firstame WITH mfirstname,
3397   socsec WITH msecsec
3398   REPLACE datebirth WITH mdatebirth
3399
3400   sex WITH msex,
3401   waterank WITH mwaterank
3402   SELECT (msec)
3403
3404   RETURN .T.
3405
3406 *****
3407
3408 * delete patient record
3409 *
3410 *
3411 *****
3412
3413 Procedure: DELPATIENT
3414
3415 *****
3416 FUNCTION delpatient
3417
3418   PARAMETERS mpatientid
3419
3420   PRIVATE msec
3421   msec = SELECT()
3422
3423   save current database selection
3424
3425   SELECT (fpatient)
3426   SET ORDER TO 3
3427
3428   SEEK mpatientid
3429   msecresult = .NOT. FOUND()
3430   IF FOUND()
3431     DELETE
3432     PACK
3433   ENDIF
3434   SET ORDER TO 1
3435
3436   store index order
3437
3438   store database selection
3439
3440   SELECT (msec)
3441
3442   RETURN msecresult
3443
3444 *****
3445
3446 *****
3447
3448 *****
3449
3450 *****
3451
3452 *****
3453
3454 *****
3455
3456 *****
3457
3458 *****
3459
3460 *****
3461
3462 *****
3463
3464 *****
3465
3466 *****
3467
3468 *****
3469
3470 *****
3471
3472 *****
3473
3474 *****
3475
3476 *****
3477
3478 *****
3479
3480 *****
3481
3482 *****
3483
3484 *****
3485
3486 *****
3487
3488 *****
3489
3490 *****
3491
3492 *****
3493
3494 *****
3495
3496 *****
3497
3498 *****
3499
3500 *****
3501
3502 *****
3503
3504 *****
3505
3506 *****
3507
3508 *****
3509
3510 *****
3511
3512 *****
3513
3514 *****
3515
3516 *****
3517
3518 *****
3519
3520 *****
3521
3522 *****
3523
3524 *****
3525
3526 *****
3527
3528 *****
3529
3530 *****
3531
3532 *****
3533
3534 *****
3535
3536 *****
3537
3538 *****
3539
3540 *****
3541
3542 *****
3543
3544 *****
3545
3546 *****
3547
3548 *****
3549
3550 *****
3551
3552 *****
3553
3554 *****
3555
3556 *****
3557
3558 *****
3559
3560 *****
3561
3562 *****
3563
3564 *****
3565
3566 *****
3567
3568 *****
3569
3570 *****
3571
3572 *****
3573
3574 *****
3575
3576 *****
3577
3578 *****
3579
3580 *****
3581
3582 *****
3583
3584 *****
3585
3586 *****
3587
3588 *****
3589
3590 *****
3591
3592 *****
3593
3594 *****
3595
3596 *****
3597
3598 *****
3599
3600 *****
3601
3602 *****
3603
3604 *****
3605
3606 *****
3607
3608 *****
3609
3610 *****
3611
3612 *****
3613
3614 *****
3615
3616 *****
3617
3618 *****
3619
3620 *****
3621
3622 *****
3623
3624 *****
3625
3626 *****
3627
3628 *****
3629
3630 *****
3631
3632 *****
3633
3634 *****
3635
3636 *****
3637
3638 *****
3639
3640 *****
3641
3642 *****
3643
3644 *****
3645
3646 *****
3647
3648 *****
3649
3650 *****
3651
3652 *****
3653
3654 *****
3655
3656 *****
3657
3658 *****
3659
3660 *****
3661
3662 *****
3663
3664 *****
3665
3666 *****
3667
3668 *****
3669
3670 *****
3671
3672 *****
3673
3674 *****
3675
3676 *****
3677
3678 *****
3679
3680 *****
3681
3682 *****
3683
3684 *****
3685
3686 *****
3687
3688 *****
3689
3690 *****
3691
3692 *****
3693
3694 *****
3695
3696 *****
3697
3698 *****
3699
3700 *****
3701
3702 *****
3703
3704 *****
3705
3706 *****
3707
3708 *****
3709
3710 *****
3711
3712 *****
3713
3714 *****
3715
3716 *****
3717
3718 *****
3719
3720 *****
3721
3722 *****
3723
3724 *****
3725
3726 *****
3727
3728 *****
3729
3730 *****
3731
3732 *****
3733
3734 *****
3735
3736 *****
3737
3738 *****
3739
3740 *****
3741
3742 *****
3743
3744 *****
3745
3746 *****
3747
3748 *****
3749
3750 *****
3751
3752 *****
3753
3754 *****
3755
3756 *****
3757
3758 *****
3759
3760 *****
3761
3762 *****
3763
3764 *****
3765
3766 *****
3767
3768 *****
3769
3770 *****
3771
3772 *****
3773
3774 *****
3775
3776 *****
3777
3778 *****
3779
3780 *****
3781
3782 *****
3783
3784 *****
3785
3786 *****
3787
3788 *****
3789
3790 *****
3791
3792 *****
3793
3794 *****
3795
3796 *****
3797
3798 *****
3799
3800 *****
3801
3802 *****
3803
3804 *****
3805
3806 *****
3807
3808 *****
3809
3810 *****
3811
3812 *****
3813
3814 *****
3815
3816 *****
3817
3818 *****
3819
3820 *****
3821
3822 *****
3823
3824 *****
3825
3826 *****
3827
3828 *****
3829
3830 *****
3831
3832 *****
3833
3834 *****
3835
3836 *****
3837
3838 *****
3839
3840 *****
3841
3842 *****
3843
3844 *****
3845
3846 *****
3847
3848 *****
3849
3850 *****
3851
3852 *****
3853
3854 *****
3855
3856 *****
3857
3858 *****
3859
3860 *****
3861
3862 *****
3863
3864 *****
3865
3866 *****
3867
3868 *****
3869
3870 *****
3871
3872 *****
3873
3874 *****
3875
3876 *****
3877
3878 *****
3879
3880 *****
3881
3882 *****
3883
3884 *****
3885
3886 *****
3887
3888 *****
3889
3890 *****
3891
3892 *****
3893
3894 *****
3895
3896 *****
3897
3898 *****
3899
3900 *****
3901
3902 *****
3903
3904 *****
3905
3906 *****
3907
3908 *****
3909
3910 *****
3911
3912 *****
3913
3914 *****
3915
3916 *****
3917
3918 *****
3919
3920 *****
3921
3922 *****
3923
3924 *****
3925
3926 *****
3927
3928 *****
3929
3930 *****
3931
3932 *****
3933
3934 *****
3935
3936 *****
3937
3938 *****
3939
3940 *****
3941
3942 *****
3943
3944 *****
3945
3946 *****
3947
3948 *****
3949
3950 *****
3951
3952 *****
3953
3954 *****
3955
3956 *****
3957
3958 *****
3959
3960 *****
3961
3962 *****
3963
3964 *****
3965
3966 *****
3967
3968 *****
3969
3970 *****
3971
3972 *****
3973
3974 *****
3975
3976 *****
3977
3978 *****
3979
3980 *****
3981
3982 *****
3983
3984 *****
3985
3986 *****
3987
3988 *****
3989
3990 *****
3991
3992 *****
3993
3994 *****
3995
3996 *****
3997
3998 *****
3999
4000 *****

```

```

3422 *****
3423
3424 *****
3425
3426 *****
3427
3428 *****
3429
3430 *****
3431
3432 *****
3433
3434 *****
3435
3436 *****
3437
3438 *****
3439
3440 *****
3441
3442 *****
3443
3444 *****
3445
3446 *****
3447
3448 *****
3449
3450 *****
3451
3452 *****
3453
3454 *****
3455
3456 *****
3457
3458 *****
3459
3460 *****
3461
3462 *****
3463
3464 *****
3465
3466 *****
3467
3468 *****
3469
3470 *****
3471
3472 *****
3473
3474 *****
3475
3476 *****
3477
3478 *****
3479
3480 *****
3481
3482 *****
3483
3484 *****
3485
3486 *****
3487
3488 *****
3489
3490 *****
3491
3492 *****
3493
3494 *****
3495
3496 *****
3497
3498 *****
3499
3500 *****
3501
3502 *****
3503
3504 *****
3505
3506 *****
3507
3508 *****
3509
3510 *****
3511
3512 *****
3513
3514 *****
3515
3516 *****
3517
3518 *****
3519
3520 *****
3521
3522 *****
3523
3524 *****
3525
3526 *****
3527
3528 *****
3529
3530 *****
3531
3532 *****
3533
3534 *****
3535
3536 *****
3537
3538 *****
3539
3540 *****
3541
3542 *****
3543
3544 *****
3545
3546 *****
3547
3548 *****
3549
3550 *****
3551
3552 *****
3553
3554 *****
3555
3556 *****
3557
3558 *****
3559
3560 *****
3561
3562 *****
3563
3564 *****
3565
3566 *****
3567
3568 *****
3569
3570 *****
3571
3572 *****
3573
3574 *****
3575
3576 *****
3577
3578 *****
3579
3580 *****
3581
3582 *****
3583
3584 *****
3585
3586 *****
3587
3588 *****
3589
3590 *****
3591
3592 *****
3593
3594 *****
3595
3596 *****
3597
3598 *****
3599
3600 *****
3601
3602 *****
3603
3604 *****
3605
3606 *****
3607
3608 *****
3609
3610 *****
3611
3612 *****
3613
3614 *****
3615
3616 *****
3617
3618 *****
3619
3620 *****
3621
3622 *****
3623
3624 *****
3625
3626 *****
3627
3628 *****
3629
3630 *****
3631
3632 *****
3633
3634 *****
3635
3636 *****
3637
3638 *****
3639
3640 *****
3641
3642 *****
3643
3644 *****
3645
3646 *****
3647
3648 *****
3649
3650 *****
3651
3652 *****
3653
3654 *****
3655
3656 *****
3657
3658 *****
3659
3660 *****
3661
3662 *****
3663
3664 *****
3665
3666 *****
3667
3668 *****
3669
3670 *****
3671
3672 *****
3673
3674 *****
3675
3676 *****
3677
3678 *****
3679
3680 *****
3681
3682 *****
3683
3684 *****
3685
3686 *****
3687
3688 *****
3689
3690 *****
3691
3692 *****
3693
3694 *****
3695
3696 *****
3697
3698 *****
3699
3700 *****
3701
3702 *****
3703
3704 *****
3705
3706 *****
3707
3708 *****
3709
3710 *****
3711
3712 *****
3713
3714 *****
3715
3716 *****
3717
3718 *****
3719
3720 *****
3721
3722 *****
3723
3724 *****
3725
3726 *****
3727
3728 *****
3729
3730 *****
3731
3732 *****
3733
3734 *****
3735
3736 *****
3737
3738 *****
3739
3740 *****
3741
3742 *****
3743
3744 *****
3745
3746 *****
3747
3748 *****
3749
3750 *****
3751
3752 *****
3753
3754 *****
3755
3756 *****
3757
3758 *****
3759
3760 *****
3761
3762 *****
3763
3764 *****
3765
3766 *****
3767
3768 *****
3769
3770 *****
3771
3772 *****
3773
3774 *****
3775
3776 *****
3777
3778 *****
3779
3780 *****
3781
3782 *****
3783
3784 *****
3785
3786 *****
3787
3788 *****
3789
3790 *****
3791
3792 *****
3793
3794 *****
3795
3796 *****
3797
3798 *****
3799
3800 *****
3801
3802 *****
3803
3804 *****
3805
3806 *****
3807
3808 *****
3809
3810 *****
3811
3812 *****
3813
3814 *****
3815
3816 *****
3817
3818 *****
3819
3820 *****
3821
3822 *****
3823
3824 *****
3825
3826 *****
3827
3828 *****
3829
3830 *****
3831
3832 *****
3833
3834 *****
3835
3836 *****
3837
3838 *****
3839
3840 *****
3841
3842 *****
3843
3844 *****
3845
3846 *****
3847
3848 *****
3849
3850 *****
3851
3852 *****
3853
3854 *****
3855
3856 *****
3857
3858 *****
3859
3860 *****
3861
3862 *****
3863
3864 *****
3865
3866 *****
3867
3868 *****
3869
3870 *****
3871
3872 *****
3873
3874 *****
3875
3876 *****
3877
3878 *****
3879
3880 *****
3881
3882 *****
3883
3884 *****
3885
3886 *****
3887
3888 *****
3889
3890 *****
3891
3892 *****
3893
3894 *****
3895
3896 *****
3897
3898 *****
3899
3900 *****
3901
3902 *****
3903
3904 *****
3905
3906 *****
3907
3908 *****
3909
3910 *****
3911
3912 *****
3913
3914 *****
3915
3916 *****
3917
3918 *****
3919
3920 *****
3921
3922 *****
3923
3924 *****
3925
3926 *****
3927
3928 *****
3929
3930 *****
3931
3932 *****
3933
3934 *****
3935
3936 *****
3937
3938 *****
3939
3940 *****
3941
3942 *****
3943
3944 *****
3945
3946 *****
3947
3948 *****
3949
3950 *****
3951
3952 *****
3953
3954 *****
3955
3956 *****
3957
3958 *****
3959
3960 *****
3961
3962 *****
3963
3964 *****
3965
3966 *****
3967
3968 *****
3969
3970 *****
3971
3972 *****
3973
3974 *****
3975
3976 *****
3977
3978 *****
3979
3980 *****
3981
3982 *****
3983
3984 *****
3985
3986 *****
3987
3988 *****
3989
3990 *****
3991
3992 *****
3993
3994 *****
3995
3996 *****
3997
3998 *****
3999
4000 *****

```

```

3478  > restore database selection          SELECT (msel)      && r
3479  <----- RETURN mresult
3480  *-----
3481  *-----
3482  * size patient list
3483  *-----
3484  *-----
3485  *-----
3493  *****
3494  Function: PATCOUNT()
3495  Called by: GETPATIENT
3496  *****
3497  *****
3498  *****
3499  *****
3500  *****
3501  *****
3502  *****
3503  *****
3504  *****
3505  *****
3506  *****
3516  *****
3517  *****
3518  *****
3519  *****
3520  *****
3521  *****
3522  *****
3523  *****
3524  *****
3525  *****
3526  *****
3527  *****
3528  *****
3529  *****
3530  *****
3531  *****
3532  *****

```

```

3533  >-----
3534  >-----
3535  >-----
3536  >-----
3537  >-----
3538  >-----
3539  >-----
3540  >-----
3541  >-----
3542  >-----
3543  >-----
3544  >-----
3545  >-----
3546  >-----
3547  >-----
3548  >-----
3549  >-----
3550  >-----
3551  >-----
3552  >-----
3553  >-----
3554  >-----
3555  >-----
3556  >-----
3557  >-----
3558  >-----
3559  >-----
3560  >-----
3561  >-----
3562  >-----
3563  >-----
3564  >-----
3572  >-----
3573  >-----
3574  >-----
3575  >-----
3576  >-----
3577  >-----
3578  >-----
3579  >-----
3580  >-----

```

```

3581 * open rule file
3582 *
3583 *
3584 *
3601 *****
3602 Procedure: OPENRULE
3603
3604 Called by: OPENFILES()
3605           (function in U.PRG)
3606
3607 Uses: PREMISE.DBF
3608       : ACTION.DBF
3609       : RULE.DBF
3610
3611 Indexes: PREMISE.IDX
3612          : ACTION.IDX
3613          : SALIENCE.IDX
3614          : RULE.IDX
3615
3616 *****
3617 FUNCTION OPENRULE
3618
3619 SELECT 0
3620 USE premise INDEX premise
3621
3622 ** list of rule premises
3623 ** dictionary object associated
3624 ** list of rule actions
3625 ** list of rules
3626 ** select rule segment
3627 ** SET RELATION TO premise INTO pr
3628 ** emise, action INTO action ** associated clauses ** rwd07
3629 RETURN .T.
3630
3631 * close rule file
3632 *
3633 *
3634 *
3635 *****
3636 Procedure: CLOSERULE
3637
3638 Called by: FLUSHFILES()
3639           (function in U.PRG)
3640
3641 *****
3642 FUNCTION CLOSERULE
3643
3644 PRIVATE msel
3645 msel = SELECT()
3646
3647 ** save current database selection
3648
3649 SELECT rule
3650 USE SELECT premise
3651 USE
3652 SELECT action
3653 USE
3654
3655 *****
3656 ** select original database
3657 RETURN mruletext
3658
3659 * data base functions for patient
3660 *
3661 *
3662 *
3663 *
3664 *
3665 *
3666 *
3667 *
3668 *
3669 *
3670 *
3671 *
3672 *
3673 *
3674 *
3675 *
3676 *
3677 *
3678 *
3679 *
3680 *
3681 *
3682 *
3683 *
3684 *
3685 *
3686 *
3687 *
3688 *
3689 *
3690 *
3691 *
3692 *
3693 *
3694 *
3695 *
3696 *
3697 *
3698 *
3699 *
3700 *
3701 *
3702 *
3703 *
3704 *
3705 *
3706 *
3707 *
3708 *
3709 *
3710 *
3711 *
3712 *
3713 *
3714 *
3715 *
3716 *
3717 *
3718 *
3719 *
3720 *
3721 *
3722 *
3723 *
3724 *
3725 *
3726 *
3727 *
3728 *
3729 *
3730 *
3731 *
3732 *
3733 *
3734 *
3735 *
3736 *
3737 *
3738 *
3739 *
3740 *
3741 *
3742 *
3743 *
3744 *
3745 *
3746 *
3747 *
3748 *
3749 *
3750 *
3751 *
3752 *
3753 *
3754 *
3755 *
3756 *
3757 *
3758 *
3759 *
3760 *
3761 *
3762 *
3763 *
3764 *
3765 *
3766 *
3767 *
3768 *
3769 *
3770 *
3771 *
3772 *
3773 *
3774 *
3775 *
3776 *
3777 *
3778 *
3779 *
3780 *
3781 *
3782 *
3783 *
3784 *
3785 *
3786 *
3787 *
3788 *
3789 *
3790 *
3791 *
3792 *
3793 *
3794 *
3795 *
3796 *
3797 *
3798 *
3799 *
3800 *
3801 *
3802 *
3803 *
3804 *
3805 *
3806 *
3807 *
3808 *
3809 *
3810 *
3811 *
3812 *
3813 *
3814 *
3815 *
3816 *
3817 *
3818 *
3819 *
3820 *
3821 *
3822 *
3823 *
3824 *
3825 *
3826 *
3827 *
3828 *
3829 *
3830 *
3831 *
3832 *
3833 *
3834 *
3835 *
3836 *
3837 *
3838 *
3839 *
3840 *
3841 *
3842 *
3843 *
3844 *
3845 *
3846 *
3847 *
3848 *
3849 *
3850 *
3851 *
3852 *
3853 *
3854 *
3855 *
3856 *
3857 *
3858 *
3859 *
3860 *
3861 *
3862 *
3863 *
3864 *
3865 *
3866 *
3867 *
3868 *
3869 *
3870 *
3871 *
3872 *
3873 *
3874 *
3875 *
3876 *
3877 *
3878 *
3879 *
3880 *
3881 *
3882 *
3883 *
3884 *
3885 *
3886 *
3887 *
3888 *
3889 *
3890 *
3891 *
3892 *
3893 *
3894 *
3895 *
3896 *
3897 *
3898 *
3899 *
3900 *
3901 *
3902 *
3903 *
3904 *
3905 *
3906 *
3907 *
3908 *
3909 *
3910 *
3911 *
3912 *
3913 *
3914 *
3915 *
3916 *
3917 *
3918 *
3919 *
3920 *
3921 *
3922 *
3923 *
3924 *
3925 *
3926 *
3927 *
3928 *
3929 *
3930 *
3931 *
3932 *
3933 *
3934 *
3935 *
3936 *
3937 *
3938 *
3939 *
3940 *
3941 *
3942 *
3943 *
3944 *
3945 *
3946 *
3947 *
3948 *
3949 *
3950 *
3951 *
3952 *
3953 *
3954 *
3955 *
3956 *
3957 *
3958 *
3959 *
3960 *
3961 *
3962 *
3963 *
3964 *
3965 *
3966 *
3967 *
3968 *
3969 *
3970 *
3971 *
3972 *
3973 *
3974 *
3975 *
3976 *
3977 *
3978 *
3979 *
3980 *
3981 *
3982 *
3983 *
3984 *
3985 *
3986 *
3987 *
3988 *
3989 *
3990 *
3991 *
3992 *
3993 *
3994 *
3995 *
3996 *
3997 *
3998 *
3999 *
4000 *

```

```

3636 *
3637 *
3638 *
3639 *
3640 *
3641 *
3642 *
3643 *
3644 *
3645 *
3646 *
3647 *
3648 *
3649 *
3650 *
3651 *
3652 *
3653 *
3654 *
3655 *
3656 *
3657 *
3658 *
3659 *
3660 *
3661 *
3662 *
3663 *
3664 *
3665 *
3666 *
3667 *
3668 *
3669 *
3670 *
3671 *
3672 *
3673 *
3674 *
3675 *
3676 *
3677 *
3678 *
3679 *
3680 *
3681 *
3682 *
3683 *
3684 *
3685 *
3686 *
3687 *
3688 *
3689 *
3690 *
3691 *
3692 *
3693 *
3694 *
3695 *
3696 *
3697 *
3698 *
3699 *
3700 *
3701 *
3702 *
3703 *
3704 *
3705 *
3706 *
3707 *
3708 *
3709 *
3710 *
3711 *
3712 *
3713 *
3714 *
3715 *
3716 *
3717 *
3718 *
3719 *
3720 *
3721 *
3722 *
3723 *
3724 *
3725 *
3726 *
3727 *
3728 *
3729 *
3730 *
3731 *
3732 *
3733 *
3734 *
3735 *
3736 *
3737 *
3738 *
3739 *
3740 *
3741 *
3742 *
3743 *
3744 *
3745 *
3746 *
3747 *
3748 *
3749 *
3750 *
3751 *
3752 *
3753 *
3754 *
3755 *
3756 *
3757 *
3758 *
3759 *
3760 *
3761 *
3762 *
3763 *
3764 *
3765 *
3766 *
3767 *
3768 *
3769 *
3770 *
3771 *
3772 *
3773 *
3774 *
3775 *
3776 *
3777 *
3778 *
3779 *
3780 *
3781 *
3782 *
3783 *
3784 *
3785 *
3786 *
3787 *
3788 *
3789 *
3790 *
3791 *
3792 *
3793 *
3794 *
3795 *
3796 *
3797 *
3798 *
3799 *
3800 *
3801 *
3802 *
3803 *
3804 *
3805 *
3806 *
3807 *
3808 *
3809 *
3810 *
3811 *
3812 *
3813 *
3814 *
3815 *
3816 *
3817 *
3818 *
3819 *
3820 *
3821 *
3822 *
3823 *
3824 *
3825 *
3826 *
3827 *
3828 *
3829 *
3830 *
3831 *
3832 *
3833 *
3834 *
3835 *
3836 *
3837 *
3838 *
3839 *
3840 *
3841 *
3842 *
3843 *
3844 *
3845 *
3846 *
3847 *
3848 *
3849 *
3850 *
3851 *
3852 *
3853 *
3854 *
3855 *
3856 *
3857 *
3858 *
3859 *
3860 *
3861 *
3862 *
3863 *
3864 *
3865 *
3866 *
3867 *
3868 *
3869 *
3870 *
3871 *
3872 *
3873 *
3874 *
3875 *
3876 *
3877 *
3878 *
3879 *
3880 *
3881 *
3882 *
3883 *
3884 *
3885 *
3886 *
3887 *
3888 *
3889 *
3890 *
3891 *
3892 *
3893 *
3894 *
3895 *
3896 *
3897 *
3898 *
3899 *
3900 *
3901 *
3902 *
3903 *
3904 *
3905 *
3906 *
3907 *
3908 *
3909 *
3910 *
3911 *
3912 *
3913 *
3914 *
3915 *
3916 *
3917 *
3918 *
3919 *
3920 *
3921 *
3922 *
3923 *
3924 *
3925 *
3926 *
3927 *
3928 *
3929 *
3930 *
3931 *
3932 *
3933 *
3934 *
3935 *
3936 *
3937 *
3938 *
3939 *
3940 *
3941 *
3942 *
3943 *
3944 *
3945 *
3946 *
3947 *
3948 *
3949 *
3950 *
3951 *
3952 *
3953 *
3954 *
3955 *
3956 *
3957 *
3958 *
3959 *
3960 *
3961 *
3962 *
3963 *
3964 *
3965 *
3966 *
3967 *
3968 *
3969 *
3970 *
3971 *
3972 *
3973 *
3974 *
3975 *
3976 *
3977 *
3978 *
3979 *
3980 *
3981 *
3982 *
3983 *
3984 *
3985 *
3986 *
3987 *
3988 *
3989 *
3990 *
3991 *
3992 *
3993 *
3994 *
3995 *
3996 *
3997 *
3998 *
3999 *
4000 *

```

```

3690      EXTERNAL ARRAY mcond, mval
3691      **
3692      PRIVATE i
3693      PRIVATE mitem
3694      PRIVATE mtype
3695      PRIVATE mid
3696      PRIVATE mq
3697      PRIVATE nrecs
3698      **
3699      ** have existing records
3700      **
3701      ** try to g
3702      **
3703      PRIVATE msel
3704      msel = SELECT()
3705      **
3706      ** find existing transactions if a
3707      **
3708      **
3709      **
3710      **
3711      **
3712      **
3713      **
3714      **
3715      **
3716      **
3717      **
3718      **
3719      **
3720      **
3721      **
3722      **
3723      **
3724      **
3725      **
3726      **
3727      **
3728      **
3729      **
3730      **
3731      **
3732      **
3733      **
3734      **
3735      **
3736      **

```

U. ACT 10-14-92 10:51a

```

3737      **
3738      **
3739      **
3740      **
3741      **
3742      **
3743      **
3744      **
3745      **
3746      **
3747      **
3748      **
3749      **
3750      **
3751      **
3752      **
3753      **
3754      **
3755      **
3756      **
3757      **
3758      **
3759      **
3760      **
3761      **
3762      **
3763      **
3764      **
3765      **
3766      **
3767      **
3768      **
3769      **
3770      **
3771      **
3772      **
3773      **
3774      **
3775      **
3776      **
3777      **
3778      **
3779      **
3780      **
3781      **
3782      **
3783      **
3784      **
3785      **
3786      **

```

Page 33 of 47

```

*****
3844 |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3845 |>>>|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3846 |>>>|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3847 |>>>|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3848 |>>>|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
*****
Function: GETSTATION()
Called by: INIENCOUNT()      (function in DEMO.PRG)
Uses: SYSTEM.DBF
*****
FUNCTION GETSTATION
PARAMETERS mstation, mdept
* get station details
PRIVATE msal
msal = SELECT()
SELECT 0
USE SYSTEM
mstation = station
mdept = department
USE
SELECT (msal)
RETURN .T.
*****
>>> save current database selection
>>> close disease file
>>> reset original database
*****
3870 |-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3871 |>>>|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
3872 |>>>|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
*****
>>> load system options
>>>
*****
Procedure: GETOPTS
Called by: SYSINIT()      (function in DEMO.PRG)
Uses: SYSTEM.DBF
*****
FUNCTION getopts
PARAMETERS ls_debug, ls_conf,
msal = SELECT()
SELECT 0
USE SYSTEM
msyswarn = syswarning
msprivate = privacyact
msdiscalm = syadiscsl
mpass = ALLTRIM(password)
ls_debug = DEBUG
*****
>>> system warnign text
>>> system privacy act text
>>> system disclaiming text
>>> system password
>>> system debug mode

```

```

3897 isplay confidence factors      is_conf = CONF      ** d
3898 isplay rules fired            is_rule = showrule     ** d
3899 isplay user mode              is_novice = novice     ** n
3900 isplay help edit mode         is_edit = edithelp    ** e
3901
3902 USE
3903 RETURN .T.
3904
3905 *****
3906 Procedure: NEXTPAT
3907 Called by: ADOPAT()
3908 Uses: SYSTEM.DBF
3909 *****
3910 FUNCTION nextpat
3911 PARAMETERS inc
3912 PRIVATE msel
3913 msel = SELECT()
3914
3915 SELECT 0
3916 USE SYSTEM
3917 IF TYPE("inc") = "L", inc,
3918
3919 *****
3920 save current database selection
3921
3922 *****
3923
3924 *****
3925
3926 *****
3927
3928 *****
3929
3930 *****
3931
3932 *****
3933
3934 *****
3935
3936 *****
3937
3938 *****
3939
3940 *****
3941
3942 *****
3943
3944 *****
3945
3946 *****
3947
3948 *****
3949
3950 *****
3951
3952 *****
3953
3954 *****
3955
3956 *****
3957
3958 *****
3959
3960 *****
3961
3962 *****
3963
3964 *****
3965
3966 *****
3967
3968 *****
3969
3970 *****
3971
3972 *****
3973
3974 *****
3975
3976 *****
3977
3978 *****
3979
3980 *****
3981
3982 *****
3983
3984 *****
3985
3986 *****
3987
3988 *****
3989
3990 *****
3991
3992 *****
3993
3994 *****
3995
3996 *****
3997
3998 *****
3999
4000 *****
4001
4002 *****
4003
4004 *****
4005
4006 *****
4007
4008 *****
4009
4010 *****
4011
4012 *****
4013
4014 *****
4015
4016 *****
4017
4018 *****
4019
4020 *****
4021
4022 *****
4023
4024 *****
4025
4026 *****
4027
4028 *****
4029
4030 *****
4031
4032 *****
4033
4034 *****
4035
4036 *****
4037
4038 *****
4039
4040 *****
4041
4042 *****
4043
4044 *****
4045
4046 *****
4047
4048 *****
4049
4050 *****
4051
4052 *****
4053
4054 *****
4055
4056 *****
4057
4058 *****
4059
4060 *****
4061
4062 *****
4063
4064 *****
4065
4066 *****
4067
4068 *****
4069
4070 *****
4071
4072 *****
4073
4074 *****
4075
4076 *****
4077
4078 *****
4079
4080 *****
4081
4082 *****
4083
4084 *****
4085
4086 *****
4087
4088 *****
4089
4090 *****
4091
4092 *****
4093
4094 *****
4095
4096 *****
4097
4098 *****
4099
4100 *****
4101
4102 *****
4103
4104 *****
4105
4106 *****
4107
4108 *****
4109
4110 *****
4111
4112 *****
4113
4114 *****
4115
4116 *****
4117
4118 *****
4119
4120 *****
4121
4122 *****
4123
4124 *****
4125
4126 *****
4127
4128 *****
4129
4130 *****
4131
4132 *****
4133
4134 *****
4135
4136 *****
4137
4138 *****
4139
4140 *****
4141
4142 *****
4143
4144 *****
4145
4146 *****
4147
4148 *****
4149
4150 *****
4151
4152 *****
4153
4154 *****
4155
4156 *****
4157
4158 *****
4159
4160 *****
4161
4162 *****
4163
4164 *****
4165
4166 *****
4167
4168 *****
4169
4170 *****
4171
4172 *****
4173
4174 *****
4175
4176 *****
4177
4178 *****
4179
4180 *****
4181
4182 *****
4183
4184 *****
4185
4186 *****
4187
4188 *****
4189
4190 *****
4191
4192 *****
4193
4194 *****
4195
4196 *****
4197
4198 *****
4199
4200 *****
4201
4202 *****
4203
4204 *****
4205
4206 *****
4207
4208 *****
4209
4210 *****
4211
4212 *****
4213
4214 *****
4215
4216 *****
4217
4218 *****
4219
4220 *****
4221
4222 *****
4223
4224 *****
4225
4226 *****
4227
4228 *****
4229
4230 *****
4231
4232 *****
4233
4234 *****
4235
4236 *****
4237
4238 *****
4239
4240 *****
4241
4242 *****
4243
4244 *****
4245
4246 *****
4247
4248 *****
4249
4250 *****
4251
4252 *****
4253
4254 *****
4255
4256 *****
4257
4258 *****
4259
4260 *****
4261
4262 *****
4263
4264 *****
4265
4266 *****
4267
4268 *****
4269
4270 *****
4271
4272 *****
4273
4274 *****
4275
4276 *****
4277
4278 *****
4279
4280 *****
4281
4282 *****
4283
4284 *****
4285
4286 *****
4287
4288 *****
4289
4290 *****
4291
4292 *****
4293
4294 *****
4295
4296 *****
4297
4298 *****
4299
4300 *****
4301
4302 *****
4303
4304 *****
4305
4306 *****
4307
4308 *****
4309
4310 *****
4311
4312 *****
4313
4314 *****
4315
4316 *****
4317
4318 *****
4319
4320 *****
4321
4322 *****
4323
4324 *****
4325
4326 *****
4327
4328 *****
4329
4330 *****
4331
4332 *****
4333
4334 *****
4335
4336 *****
4337
4338 *****
4339
4340 *****
4341
4342 *****
4343
4344 *****
4345
4346 *****
4347
4348 *****
4349
4350 *****
4351
4352 *****
4353
4354 *****
4355
4356 *****
4357
4358 *****
4359
4360 *****
4361
4362 *****
4363
4364 *****
4365
4366 *****
4367
4368 *****
4369
4370 *****
4371
4372 *****
4373
4374 *****
4375
4376 *****
4377
4378 *****
4379
4380 *****
4381
4382 *****
4383
4384 *****
4385
4386 *****
4387
4388 *****
4389
4390 *****
4391
4392 *****
4393
4394 *****
4395
4396 *****
4397
4398 *****
4399
4400 *****
4401
4402 *****
4403
4404 *****
4405
4406 *****
4407
4408 *****
4409
4410 *****
4411
4412 *****
4413
4414 *****
4415
4416 *****
4417
4418 *****
4419
4420 *****
4421
4422 *****
4423
4424 *****
4425
4426 *****
4427
4428 *****
4429
4430 *****
4431
4432 *****
4433
4434 *****
4435
4436 *****
4437
4438 *****
4439
4440 *****
4441
4442 *****
4443
4444 *****
4445
4446 *****
4447
4448 *****
4449
4450 *****
4451
4452 *****
4453
4454 *****
4455
4456 *****
4457
4458 *****
4459
4460 *****
4461
4462 *****
4463
4464 *****
4465
4466 *****
4467
4468 *****
4469
4470 *****
4471
4472 *****
4473
4474 *****
4475
4476 *****
4477
4478 *****
4479
4480 *****
4481
4482 *****
4483
4484 *****
4485
4486 *****
4487
4488 *****
4489
4490 *****
4491
4492 *****
4493
4494 *****
4495
4496 *****
4497
4498 *****
4499
4500 *****
4501
4502 *****
4503
4504 *****
4505
4506 *****
4507
4508 *****
4509
4510 *****
4511
4512 *****
4513
4514 *****
4515
4516 *****
4517
4518 *****
4519
4520 *****
4521
4522 *****
4523
4524 *****
4525
4526 *****
4527
4528 *****
4529
4530 *****
4531
4532 *****
4533
4534 *****
4535
4536 *****
4537
4538 *****
4539
4540 *****
4541
4542 *****
4543
4544 *****
4545
4546 *****
4547
4548 *****
4549
4550 *****
4551
4552 *****
4553
4554 *****
4555
4556 *****
4557
4558 *****
4559
4560 *****
4561
4562 *****
4563
4564 *****
4565
4566 *****
4567
4568 *****
4569
4570 *****
4571
4572 *****
4573
4574 *****
4575
4576 *****
4577
4578 *****
4579
4580 *****
4581
4582 *****
4583
4584 *****
4585
4586 *****
4587
4588 *****
4589
4590 *****
4591
4592 *****
4593
4594 *****
4595
4596 *****
4597
4598 *****
4599
4600 *****
4601
4602 *****
4603
4604 *****
4605
4606 *****
4607
4608 *****
4609
4610 *****
4611
4612 *****
4613
4614 *****
4615
4616 *****
4617
4618 *****
4619
4620 *****
4621
4622 *****
4623
4624 *****
4625
4626 *****
4627
4628 *****
4629
4630 *****
4631
4632 *****
4633
4634 *****
4635
4636 *****
4637
4638 *****
4639
4640 *****
4641
4642 *****
4643
4644 *****
4645
4646 *****
4647
4648 *****
4649
4650 *****
4651
4652 *****
4653
4654 *****
4655
4656 *****
4657
4658 *****
4659
4660 *****
4661
4662 *****
4663
4664 *****
4665
4666 *****
4667
4668 *****
4669
4670 *****
4671
4672 *****
4673
4674 *****
4675
4676 *****
4677
4678 *****
4679
4680 *****
4681
4682 *****
4683
4684 *****
4685
4686 *****
4687
4688 *****
4689
4690 *****
4691
4692 *****
4693
4694 *****
4695
4696 *****
4697
4698 *****
4699
4700 *****
4701
4702 *****
4703
4704 *****
4705
4706 *****
4707
4708 *****
4709
4710 *****
4711
4712 *****
4713
4714 *****
4715
4716 *****
4717
4718 *****
4719
4720 *****
4721
4722 *****
4723
4724 *****
4725
4726 *****
4727
4728 *****
4729
4730 *****
4731
4732 *****
4733
4734 *****
4735
4736 *****
4737
4738 *****
4739
4740 *****
4741
4742 *****
4743
4744 *****
4745
4746 *****
4747
4748 *****
4749
4750 *****
4751
4752 *****
4753
4754 *****
4755
4756 *****
4757
4758 *****
4759
4760 *****
4761
4762 *****
4763
4764 *****
4765
4766 *****
4767
4768 *****
4769
4770 *****
4771
4772 *****
4773
4774 *****
4775
4776 *****
4777
4778 *****
4779
4780 *****
4781
4782 *****
4783
4784 *****
4785
4786 *****
4787
4788 *****
4789
4790 *****
4791
4792 *****
4793
4794 *****
4795
4796 *****
4797
4798 *****
4799
4800 *****
4801
4802 *****
4803
4804 *****
4805
4806 *****
4807
4808 *****
4809
4810 *****
4811
4812 *****
4813
4814 *****
4815
4816 *****
4817
4818 *****
4819
4820 *****
4821
4822 *****
4823
4824 *****
4825
4826 *****
4827
4828 *****
4829
4830 *****
4831
4832 *****
4833
4834 *****
4835
4836 *****
4837
4838 *****
4839
4840 *****
4841
4842 *****
4843
4844 *****
4845
4846 *****
4847
4848 *****
4849
4850 *****
4851
4852 *****
4853
4854 *****
4855
4856 *****
4857
4858 *****
4859
4860 *****
4861
4862 *****
4863
4864 *****
4865
4866 *****
4867
4868 *****
4869
4870 *****
4871
4872 *****
4873
4874 *****
4875
4876 *****
4877
4878 *****
4879
4880 *****
4881
4882 *****
4883
4884 *****
4885
4886 *****
4887
4888 *****
4889
4890 *****
4891
4892 *****
4893
4894 *****
4895
4896 *****
4897
4898 *****
4899
4900 *****
4901
4902 *****
4903
4904 *****
4905
4906 *****
4907
4908 *****
4909
4910 *****
4911
4912 *****
4913
4914 *****
4915
4916 *****
4917
4918 *****
4919
4920 *****
4921
4922 *****
4923
4924 *****
4925
4926 *****
4927
4928 *****
4929
4930 *****
4931
4932 *****
4933
4934 *****
4935
4936 *****
4937
4938 *****
4939
4940 *****
4941
4942 *****
4943
4944 *****
4945
4946 *****
4947
4948 *****
4949
4950 *****
4951
4952 *****
4953
4954 *****
4955
4956 *****
4957
4958 *****
4959
4960 *****
4961
4962 *****
4963
4964 *****
4965
4966 *****
4967
4968 *****
4969
4970 *****
4971
4972 *****
4973
4974 *****
4975
4976 *****
4977
4978 *****
4979
4980 *****
4981
4982 *****
4983
4984 *****
4985
4986 *****
4987
4988 *****
4989
4990 *****
4991
4992 *****
4993
4994 *****
4995
4996 *****
4997
4998 *****
4999
5000 *****

```



```

4234 msoc = dp(mitem, u, 1)
4235 mlast = dp(mitem, u, 2)
4236 mfirst = dp(mitem, u, 3)
4237 mmid = dp(mitem, u, 4)
4238 mpresize = ALEN(choice, 1)
4239 DIMENSION mprelist[mpresize]
4240 FOR i = 1 TO mpresize
4241   mtsn = dp(choice[i, 1], "(",
4242   mtsn = dp(mtsn, ")", 1)
4243   mprelist[i] = mtsn
4244 ENDFOR
4245 SET EXACT OFF
4246 mmsoc = .f.
4247 IF EMPTY(msoc) AND EMPTY(mlast)
4248   AND EMPTY(mfirst)
4249   AND WITHOUT CRITERIA")
4250 RETURN ""
4251 winpop = popupshow ("searching
4252   "
4253   "
4254   "
4255   "
4256   "
4257   "
4258   "
4259   "
4260   "
4261   "
4262   "
4263   "
4264   "
4265   "
4266   "
4267   "
4268   "
4269   "
4270   "
4271   "
4272   "
4273   "
4274   "
4275   "
4276   "
4277   "
4278   "
4279   "
4280   "
4281   "
4282   "
4283   "
4284   "
4285   "

```

```

4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

```

4341 tfirst = mfirst
4342 IF msmsdata
4343   IF OCCURS(" ",msn) = 0
4344     tlast = ""
4345     tfirst = ""
4346   ELSE
4347     tsn = STRTRAN(tsn, " ",
4348     )
4349   ENDIF
4350   tsn = STRTRAN(tsn, " ",
4351   )
4352   tlast = STRTRAN(tsn, " ",
4353   )
4354   tfirst = STRTRAN(tsn, " ",
4355   )
4356   tlast = ALLTRIM(tlast)
4357   tfirst = ALLTRIM(tfirst)
4358   SELECT ssn, name, crwdob, crws
4359   FROM (f mkey);
4360   INTO CURSOR upat2;
4361   WHERE ssn LIKE tsn;
4362   AND name LIKE (UPPER(tl
4363   )
4364   )
4365   )
4366   )
4367   )
4368   )
4369   )
4370   )
4371   )
4372   )
4373   )
4374   )
4375   )
4376   )
4377   )
4378   )
4379   )
4380   )
4381   )
4382   )
4383   )
4384   )
4385   )
4386   )
4387   )
4388   )
4389   )
4390   )
4391   )
4392   )
4393   )
4394   )
4395   )
4396   )
4397   )
4398   )
4399   )
4400   )
4401   )
4402   )
4403   )
4404   )
4405   )

```

Procedure: GETSAMS

```

4406 *
4407 *
4408 *
4409 *
4410 *
4411 *
4412 *
4413 *
4414 *
4415 *
4416 *
4417 *
4418 *
4419 *
4420 *
4421 *
4422 *
4423 *
4424 *
4425 *
4426 *
4427 *
4428 *
4429 *
4430 *
4431 *
4432 *
4433 *
4434 *
4435 *
4436 *
4437 *
4438 *
4439 *
4440 *
4441 *
4442 *
4443 *
4444 *
4445 *
4446 *
4447 *
4448 *
4449 *
4450 *
4451 *
4452 *
4453 *
4454 *
4455 *
4456 *

```

Procedure: GETPAT


```

4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628

IF EMPTY(mrec)
  SELECT (msel)
  RETURN
ENDIF
GO RECORD mrec
ENDIF
REPLACE id WITH mpatid
REPLACE socsec WITH mssn
REPLACE lastname WITH mlast
REPLACE firstname WITH mfirst
REPLACE middleinit WITH mmid
REPLACE datebirth WITH mdob
REPLACE sex WITH msex
REPLACE ratenrank WITH mrank
mdata = combined(mssn,mlast,mf
=> first,mmid,dtoc(mdob),msex,mrank,STR(mpatid),STR(mrec))
SELECT (msel)
RETURN mdata
*****
* Open SAMS database.
*****
=> *****
Procedure: OPENSAMS
Called by: OPENPAT
Calls: SETPATH
: FILEFIND
Uses: &F_MKEY
*****
FUNCTION opensams
PUBLIC f_mkey, f_mkeyssn, fssm
f_mkey = "mkey"
f_mkeyssn = "mkeyssn"
IF EMPTY(GETENV("SAMS"))
DO setpath WITH GETENV("SAMS")
ENDIF
mfind = filefind(f_mkey + ".d
=> btf")
IF mfind
RETURN .f.
ENDIF
IF EMPTY(fssm) OR IUSED(f_mkey
SELECT 0
USE &f_mkey AGAIN INDEX (f_
=> mkeyssn)
fssm = SELECT()
ELSE
SELECT (fssm)
fssm = SELECT()
ENDIF
RETURN .t.
*****
* Convert first letter of sent
=> ense to lower case or upper case
*****

```

```

=> *****
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828

Procedure: CASECONVERT
Called by: SFSYMP
*****
FUNCTION caseconvert
PARAMETER mstr, mcase
DO CASE
CASE mcase = "U"
mtext = UPPER(LEFT(mstr,1))
=> SUBSTR(mstr,2)
CASE mcase = "L"
mtext = LOWER(LEFT(mstr,1))
=> SUBSTR(mstr,2)
OTHERWISE
mtext = ""
ENDCASE
RETURN mtext
*****
* Get the encounter for sf600
*****
=> report
=> *****
Procedure: SFENCOUNT
Called by: M_SF600
Calls: M_PATGET
: PUSHHELP
: WPA, SPR
: POPHELP
: SFGETPAT
: SFGETENC
: POPMESSAGE()
: PATSSNK()
: LOENC
*****
*****
FUNCTION sfencount
PRIVATE mprevpat, mprevdate
mprevpat = mpatientid
mprevdate = mdateenc
mdateenc = CTOD(" / / ")
nbuttons = 1
IF mpatientid > 0
=> mpatget(mpatientid,msocse
=> c,amlastname,amfirstame,amdatebirth,amsex,amrate,amrank)
ENDIF
msn = 1 IF (EMPTY(msocsec),"
=> "msocsec)
mlast = mlastname
mfirst = mfirstname
mdob = mdatebirth
msex = msex
mrate = mrate,rank
=> pushhelp("SF600 REPORT SELECT
=> 10")

```

```

4685 DO mp4.spr
4686 *pophelp()
4687 DO CASE
4688 CASE buttons = 1
4689 *patientid = sfgetpat(masn,
4690 *least, mfirst)
4691 CASE *patientid > 0
4692 CASE *patientid > 0
4693 *patientid = sfgetenc
4694 *patientid, mdateenc)
4695 *patientid)
4696 *patientid)
4697 *patientid)
4698 *patientid)
4699 *patientid)
4700 *patientid)
4701 *patientid)
4702 *patientid)
4703 *patientid)
4704 *patientid)
4705 *patientid)
4706 *patientid)
4707 *patientid)
4708 *patientid)
4709 *patientid)
4710 *patientid)
4711 *patientid)
4712 *patientid)
4713 *patientid)
4714 *patientid)
4715 *patientid)
4716 *patientid)
4717 *patientid)
4718 *patientid)
4719 *patientid)
4720 *patientid)
4721 *patientid)
4722 *patientid)
4723 *patientid)
4724 *patientid)
4725 *patientid)
4726 *patientid)
4727 *patientid)
4728 *patientid)
4729 *patientid)
4730 *patientid)
4731 *patientid)
4732 *patientid)
4733 *patientid)
4734 *patientid)
4735 *patientid)
4736 *patientid)
4737 *patientid)
4738 *patientid)
4739 *patientid)
4740 *patientid)
4741 *patientid)
4742 *patientid)
4743 *patientid)
4744 *patientid)
4745 *patientid)
4746 *patientid)
4747 *patientid)
4748 *patientid)
4749 *patientid)
4750 *patientid)
4751 *patientid)
4752 *patientid)

```

```

4741 *patientid)
4742 *patientid)
4743 *patientid)
4744 *patientid)
4745 *patientid)
4746 *patientid)
4747 *patientid)
4748 *patientid)
4749 *patientid)
4750 *patientid)
4751 *patientid)
4752 *patientid)
4753 *patientid)
4754 *patientid)
4755 *patientid)
4756 *patientid)
4757 *patientid)
4758 *patientid)
4759 *patientid)
4760 *patientid)
4761 *patientid)
4762 *patientid)
4763 *patientid)
4764 *patientid)
4765 *patientid)
4766 *patientid)
4767 *patientid)
4768 *patientid)
4769 *patientid)
4770 *patientid)
4771 *patientid)
4772 *patientid)
4773 *patientid)
4774 *patientid)
4775 *patientid)
4776 *patientid)
4777 *patientid)
4778 *patientid)
4779 *patientid)
4780 *patientid)
4781 *patientid)
4782 *patientid)
4783 *patientid)
4784 *patientid)
4785 *patientid)
4786 *patientid)
4787 *patientid)
4788 *patientid)
4789 *patientid)
4790 *patientid)
4791 *patientid)
4792 *patientid)
4793 *patientid)

```

Procedure: SFGETENC

Called by: SFENCOUNT

Calls: ENCOUNTER()

: SFENCOUNT

: SFENCOUNT

: SFENCOUNT

: SFENCOUNT


```

4903 >> s, spatient, mdate
4904 >> ients
4905 >> ients
4906 >> ients
4907 >> ients
4908 >> ients
4909 >> ients
4910 >> ients
4911 >> ients
4912 >> ients
4913 >> ients
4914 >> ients
4915 >> ients
4916 >> ients
4917 >> ients
4918 >> ients
4919 >> ients
4920 >> save current database selection
4921 >> save current database selection
4922 >> save current database selection
4923 >> save current database selection
4924 >> save current database selection
4925 >> save current database selection
4926 >> save current database selection
4927 >> save current database selection
4928 >> save current database selection
4929 >> save current database selection
4930 >> save current database selection
4931 >> save current database selection
4932 >> save current database selection
4933 >> save current database selection
4934 >> save current database selection
4935 >> save current database selection
4936 >> save current database selection
4937 >> save current database selection
4938 >> save current database selection
4939 >> save current database selection
4940 >> save current database selection
4941 >> save current database selection
4942 >> save current database selection
4943 >> save current database selection
4944 >> save current database selection
4945 >> save current database selection
4946 >> save current database selection
4947 >> save current database selection
4948 >> save current database selection
4949 >> save current database selection
4950 >> save current database selection
4951 >> save current database selection
4952 >> save current database selection
4953 >> save current database selection
4954 >> save current database selection
4955 >> save current database selection
4956 >> save current database selection
4957 >> save current database selection
4958 >> save current database selection

```


5250	REINDEX COMPACT
5251	USE VAL INDEX VAL
5252	REINDEX COMPACT
5253	USE enum INDEX enum
5254	REINDEX COMPACT
5255	USE premise INDEX premise
5256	REINDEX COMPACT
5257	USE action INDEX action
5258	REINDEX COMPACT
5259	USE rule INDEX salience, ru
5260	as le
5261	REINDEX COMPACT
5262	USE disease INDEX disease,
5263	as disid, disease
5264	REINDEX COMPACT
5265	USE diagnose INDEX diagnose
5266	REINDEX COMPACT
5267	USE REPORT INDEX REPORT
5268	REINDEX COMPACT
5269	USE SECTION INDEX SECTION
5270	REINDEX COMPACT
5271	USE sf600tex INDEX sf600tex
5272	REINDEX COMPACT
5273	as popphide(windpop)
5275	END IF
	RETURN
	*: EOF: U.ACT

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

19
20
21
22
23
24
25
26
27
28
29

2000

38
39
40
41
42
43
44
45
46
47

25

58
59
60
61
62

38

.....

67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130

11

```

133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
179 *

```

_q7i0lh94d mbuttons VALID

Function Origin:

From Screen: W2, Record Number: 6

Variable: mbuttons

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 2

```

145 FUNCTION q7i0lh94d      && mbuttons VALID
146 #REGION 1
147 IF mbuttons = 1
148      = printenc(ALLTRIM(mprintfile))
149 ENDIF
150 RETURN .T.

```

_q7i0lh94q mchoice VALID

Function Origin:

From Screen: W2, Record Number: 7

Variable: mchoice

Called By: VALID Clause

Object Type: Radio Button

Snippet Number: 3

```

153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
179 *

```

FUNCTION q7i0lh94q && mchoice VALID

#REGION 1

IF mchoice = 1

 mprintfile = mfilename

 CURJOB = OBJNUM(mprintfile)

ELSE

 mprintfile = "PRINT"

ENDIF

SHOW GETS

RETURN .T.

*: EOF: W2.AC1

07/20/92	W3.SPR	10:01:25
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

REGION 0
REGIONAL m.curarea, m.talkstat, m.compstat
-IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
-ELSE
m.talkstat = "OFF"
-ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

-IF NOT MEXIST("w_patlist")
DEFINE WINDOW w_patlist
FROM INT((SCOL()-15)/2), INT((SCOL()-37)/2) ;
TO INT((SCOL()-15)/2)+14, INT((SCOL()-37)/2)+36 ;
TITLE "Patients on File" ;
FLOAT ;
NOCLOSE ;
SHADOW ;
DOUBLE ;
COLOR SCHEME 5
-ENDIF

```

W3 Setup Code - SECTION 2

```

REGION 1
mchoice=1

```

W3 Screen Layout

```

67 REGION 1
68 -IF WVISIBLE("w_patlist")
69 ACTIVATE WINDOW w_patlist SAME
70 -ELSE
71 ACTIVATE WINDOW w_patlist NOSHOW
72 -ENDIF
73 @ 0,0 GET mchoice ;
74 PICTURE "set" ;
75 FROM mlist ;
76 SIZE 11,34 ;
77 DEFAULT 1 ;
78 COLOR SCHEME 6
79 @ 12,6 GET mbuttons ;
80 PICTURE "g=HT Select;\?Cancel" ;
81 SIZE 1,8,5 ;
82 DEFAULT 1 ;
83
84 -IF NOT WVISIBLE("w_patlist")
85 ACTIVATE WINDOW w_patlist
86 -ENDIF
87
88 READ CYCLE MODAL
89
90 RELEASE WINDOW w_patlist
91
92 REGION 0
93 -IF m.talkstat = "ON"
94 SET TALK ON
95 -ENDIF
96 -IF m.compstat = "ON"
97 SET COMPATIBLE ON
98 -ENDIF
99 =: EOF: W3.AC1
101

```

07/28/92	W4.SPR	16:29:59
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

#REGION 0
REGIONAL m.curraea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT EXIST("w_patedit")
  DEFINE WINDOW w_patedit
  FROM INT((SROW()-23)/2), INT((SCOL()-80)/2)
  TO INT((SROW()-23)/2)+22, INT((SCOL()-80)/2)+79
  TITLE "Patient Details"
  FLOAT
  NOCLOSE
  SHADOW
  DOUBLE
  COLOR SCHEME 5
ENDIF

```

W4 Screen Layout

```

#REGION 1
IF WISIBLE("w_patedit")
  ACTIVATE WINDOW w_patedit SAME
ELSE
  ACTIVATE WINDOW w_patedit NOSHOW
ENDIF
# 1,1 SAY "Social security"
# 3,1 SAY "Last name"
# 5,1 SAY "First name"

```

```

# 7,1 SAY "Date of birth"
# 9,1 SAY "Sex"
# 11,1 SAY "Rate / rank"
# 1,18 GET msocsec
  SIZE 1,12
  DEFAULT " "
  PICTURE "999-99-9999"
  DISABLE
# 3,18 GET mlastname
  SIZE 1,27
  DEFAULT " "
  DISABLE
# 5,18 GET mfirstname
  SIZE 1,27
  DEFAULT " "
  DISABLE
# 7,18 GET mdatebirth
  SIZE 1,8
  DEFAULT ( / / )
  DISABLE
# 9,18 GET msecx
  SIZE 1,1
  DEFAULT " "
  PICTURE "9A1"
  DISABLE
# 11,18 GET mratebank
  SIZE 1,28
  DEFAULT " "
  DISABLE
# 7,66 GET mbuttons
  PICTURE "g*VT OK;New;\Cancel"
  SIZE 1,9,2
  DEFAULT " "
  VALID _pvalidd()
  PICTURE "g*HN Edit"
  SIZE 1,6,1
  DEFAULT " "
  VALID _q7q0z05ur()
# 4,66 GET mbutton
  PICTURE "g*HN Browse"
  SIZE 1,8,1
  DEFAULT " "
  VALID _q7q0z05j()
IF NOT WISIBLE("w_patedit")
  ACTIVATE WINDOW w_patedit
ENDIF
READ CYCLE
DEACTIVATE _stopread("w_patedit")
RELEASE WINDOW w_patedit
#REGION 0
IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compstat = "ON"
  SET COMPATIBLE ON
ENDIF

```

```

133 *
134 *
135 *
136 #REGION 1
137 -IF Quitting
138   RELEASE WINDOW w_patedit
139 -ENDIF
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155

```

_q7q0zd5ur	mbutton VALID
Function Origin:	
From Screen:	W4, Record Number: 15
Variable:	mbutton
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	1

```

156 FUNCTION _q7q0zd5ur 24 mbutton VALID
157 #REGION 1
158 -IF saving
159   saving = .f.
160   SHOW GETS disabled
161 -ELSE
162   saving = .t.
163   SHOW GETS enabled
164 -ENDIF
165 RETURN .t.
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180

```

_q7q0zd65j	mbutton. VALID
Function Origin:	
From Screen:	W4, Record Number: 16
Variable:	mbutton
Called By:	VALID Clause
Object Type:	Push Button
Snippet Number:	2

```

181 FUNCTION _q7q0zd65j 24 mbutton VALID
182 #REGION 1
183 DO m_patlist IN (&startdir + "\\" + "demo.prg")
184 SHOW GETS
185 RETURN .t.
186 *: EOF: W4.AC1
187

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

07/20/92	W5.SPR	10-01:21
Author's Name		
Copyright (c) 1992 Company Name		
Address		
City,	Zip	
Description: This program was automatically generated by GENSCRN.		

#REGION 0
REGIONAL m.curraarea, m.talkstat, m.compmat

IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compmat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

Window definitions

IF NOT MEXIST("w_area")
DEFINE WINDOW w_area
FROM INT((SROW()-11)/2), INT((SCOL()-46)/2) ;
TO INT((SROW()-11)/2)+10, INT((SCOL()-46)/2)+45 ;
TITLE "Complaints"
FLOAT
NOCLOSE ;
SHADOW ;
DOUBLE ;
COLOR SCHEME 5
ENDIF

W5 Setup Code - SECTION 2

#REGION 1
mchoice = 1

W5 Screen Layout

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```
#REGION 1
IF WISIBLE("w_area")
  ACTIVATE WINDOW w_area SAME
ELSE
  ACTIVATE WINDOW w_area NOSHOW
ENDIF
@ 2,32 GET mbuttons ;
  PICTURE "g+vt Select;\?Cancel" ;
  SIZE 1,8,1 ;
  DEFAULT 1 ;
@ 0,1 GET mchoice ;
  PICTURE "a&t" ;
  FROM mlist ;
  SIZE 9,29 ;
  DEFAULT 1 ;
  COLOR SCHEME 6

IF NOT WISIBLE("w_area")
  ACTIVATE WINDOW w_area
ENDIF

READ CYCLE MODAL

RELEASE WINDOW w_area

#REGION 0
IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compmat = "ON"
  SET COMPATIBLE ON
ENDIF
* ; EOF: W5.AC1
```


07/20/92	W6.SPR	10:00:49
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

REGION 0
REGIONAL i.currare, m.talkstat, m.compatat

```

```

IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF

```

```

m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT EXIST("w_num")
DEFINE WINDOW w_num ;
FROM INT((SCOL()-15)/2), INT((SCOL()-55)/2) ;
TO INT((SCOL()-15)/2)+16, INT((SCOL()-55)/2)+56 ;
TITLE "Numeric Data Entry" ;
FLOAT ;
NOCLOSE ;
SHADOW ;
DOUBLE ;
COLOR SCHEME 5
ENDIF

```

W6 Screen Layout

```

REGION 1
IF WISIBLE("w_num")
ACTIVATE WINDOW w_num SAME
ELSE
ACTIVATE WINDOW w_num WOSHON
ENDIF
@ 8,6 GET mval ;
SIZE 1,10 ;
RANGE mlo, mhi ;

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

```

```

DEFAULT 0 ;
PICTURE "BKZ"
@ 1,1 SAY mprompt ;
SIZE 3,5 ;
PICTURE "a1"
@ 6,2 TO 10,48
@ 5,23 SAY mhi ;
SIZE 1,10 ;
PICTURE "BKZ"
@ 8,30 SAY munits ;
SIZE 1,14 ;
PICTURE "a1"
@ 5,7 SAY mlo ;
SIZE 1,10 ;
PICTURE "BKZ"
@ 5,19 SAY "to"
@ 12,16 GET mbuttons ;
PICTURE "a1" OK;\Cancel" ;
SIZE 1,8,5 ;
DEFAULT 1 ;
@ 5,1 SAY "Range"
IF NOT WISIBLE("w_num")
ACTIVATE WINDOW w_num
ENDIF
READ CYCLE MODAL
RELEASE WINDOW w_num
REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compatat = "ON"
SET COMPATIBLE ON
ENDIF
* EOF : W6.AC1

```

07/20/92	W68.SPR	10:00:58
Author's Name		
Copyright (c) 1992 Company Name		
Address		
City, Zip		
Description:		
This program was automatically generated by GENSCREEN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

REGION 0
REGIONAL m.curraee, m.talkstat, m.compatat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT WEXIST("w_check")
DEFINE WINDOW w_check :
FROM INT((SROW()-17)/2), INT((SCOL()-63)/2) :
TO INT((SROW()-17)/2)+16, INT((SCOL()-63)/2)+62 :
TITLE "Single Choice Data Entry" :
FLOAT :
NOCLOSE :
SHADOW :
DOUBLE :
COLOR SCHEME 5
ENDIF

```

W68 Screen Layout

```

REGION 1
IF WVISIBLE("w_check")
ACTIVATE WINDOW w_check SAME
ELSE
ACTIVATE WINDOW w_check NOSHOW
ENDIF
@ 14,20 GET buttons ;
PICTURE "3-INT OK;\Cancel" ;
SIZE 1,8,5 ;

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

```

```

DEFAULT 1
@ 1,1 SAY prompt ;
SIZE 3,59 ;
PICTURE "a1" ;
@ 5,1 GET mchoice ;
FROM choices ;
SIZE 8,59 ;
DEFAULT 1 ;
COLOR SCHEME 6

IF NOT WVISIBLE("w_check")
ACTIVATE WINDOW w_check
ENDIF

READ CYCLE MODAL

RELEASE WINDOW w_check

REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compatat = "ON"
SET COMPATIBLE ON
ENDIF
* : EOF: W68.AC1

```

07/20/92	W6C.SPR	10:00:53
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

#REGION 0
REGIONAL m.curraea, m.talkstat, m.compatat

```

```

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT WEXIST("w_check")
  DEFINE WINDOW w_check
  FROM INT((SROW()-14)/2), INT((SCOL()-54)/2)
  TO INT((SROW()-14)/2)+13, INT((SCOL()-54)/2)+53 ;
  TITLE "Logical Data Entry" ;
  FLOAT ;
  NOCLOSE ;
  SHADOW ;
  DOUBLE ;
  COLOR SCHEME 5
ENDIF

```

W6C Screen Layout

```

#REGION 1
IF WVISIBLE("w_check")
  ACTIVATE WINDOW w_check SAME
ELSE
  ACTIVATE WINDOW w_check NOSHOW
ENDIF
@ 11,14 GET mbuttons ;
  PICTURE "RIGHT OK;\?Cancel" ;
  SIZE 1,8,5 ;

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93

```

```

DEFAULT 1
@ 1,1 SAY mprompt ;
SIZE 3,50 ;
PICTURE "gt" ;
@ 6,22 GET mchoice ;
PICTURE "g-rvm yes:No" ;
SIZE 1,7,0 ;
DEFAULT 1 ;
@ 4,13 TO 9,38
  IF NOT WVISIBLE("w_check")
    ACTIVATE WINDOW w_check
  ENDIF
  READ CYCLE MODAL
  RELEASE WINDOW w_check
#REGION 0
IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compatat = "ON"
  SET COMPATIBLE ON
ENDIF
* : EOF: W6C.AC1

```

07/20/92	W7.SPR	10:00:40
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

```

```

18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

REGION 0
REGIONAL m.curraee, m.talkstat, m.compatstat
-IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
-ELSE
m.talkstat = "OFF"
-ENDIF
m.compatstat = SET("COMPATIBLE")
SET COMPATIBLE FORPLUS

```

Window definitions

```

-IF NOT MEXIST("w_disease")
DEFINE WINDOW w_disease ;
FROM INT((SROW()-16)/2),INT((SCOL()-39)/2) ;
TO INT((SROW()-16)/2)+15,INT((SCOL()-39)/2)+38 ;
TITLE "Diseases on file" ;
FLOAT ;
NOCLOSE ;
SHADOW ;
DOUBLE ;
COLOR SCHEME 5
-ENDIF

```

W7 Setup Code - SECTION 2

```

REGION 1
mchoice = 1

```

W7 Screen Layout

W7.AC1 10-16-92 10:51a

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

```

REGION 1
-IF WVISIBLE("w_disease")
ACTIVATE WINDOW w_disease SAME
-ELSE
ACTIVATE WINDOW w_disease WSHOU
-ENDIF
a 0,0 GET echoice ;
PICTURE "a87" ;
FROM mlst ;
SIZE 11,36 ;
DEFAULT 1 ;
COLOR SCHEME 6
a 12,1 GET abuttons ;
PICTURE "a8HT Select;View;VCancel" ;
SIZE 1,8,4 ;
DEFAULT 1 ;
-IF NOT WVISIBLE("w_disease")
ACTIVATE WINDOW w_disease
-ENDIF
READ CYCLE MODAL
RELEASE WINDOW w_disease
REGION 0
-IF m.talkstat = "ON"
SET TALK ON
-ENDIF
-IF m.compatstat = "ON"
SET COMPATIBLE ON
-ENDIF
* : EOF: W7.AC1

```

Page 1 of 1

08/03/92	W7A.SPR	16:06:10
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

REGION 0
REGIONAL m.curraea, m.talkstat, m.compatat

```

```

IF SET("TALK") = "ON"
SET TALK OFF
ELSE
m.talkstat = "ON"
m.talkstat = "OFF"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT MEXIST("w_dislook")
DEFINE WINDOW w_dislook ;
FROM INT((SQW()-19)/2), INT((SCOL()-40)/2) ;
TO INT((SQW()-19)/2)+18, INT((SCOL()-40)/2)+39 ;
TITLE "Disease/Treatment Reference" ;
NOFLOAT ;
NOCLOSE ;
SHADOW ;
DOUBLE ;
COLOR SCHEME 5
ENDIF

```

W7A Setup Code - SECTION 2

```

REGION 1
mchoice = 1

```

W7A Screen Layout

```

REGION 1
IF VISIBLE("w_dislook")
ACTIVATE WINDOW w_dislook SAME
ELSE
ACTIVATE WINDOW w_dislook NOSHOW
ENDIF
8 0,1 GET mchoice ;
PICTURE "a1" ;
FROM wlist ;
SIZE 15,35 ;
DEFAULT 1 ;
VALID q7dy1112( ) ;
COLOR SCHEME 6
8 15,10 GET mbuttons ;
PICTURE "a2" \\<View>\\<Quit> ;
SIZE 1,8,4 ;
DEFAULT 1 ;
IF NOT VISIBLE("w_dislook")
ACTIVATE WINDOW w_dislook
ENDIF
READ CYCLE
RELEASE WINDOW w_dislook
REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compatat = "ON"
SET COMPATIBLE ON
ENDIF
...
_q7dy1112 mchoice VALID
Function Origin:
From Screen: W7A, Record Number: 2
Variable: mchoice
Called By: VALID Clause
Object Type: List
Snippet Number: 1
FUNCTION _q7dy1112 AS mchoice VALID
REGION 1
mbuttons = 1
* : EOF: W7A.AC1

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121

```

10/13/92	W8.PRG	09:54:30
Roy Dobbin & Hoa Ly		
Copyright (c) 1992		
Naval Health Research Center, code 22		
P.O. Box 85122		
San Diego, CA 92186-5122		
Description:		
This program was automatically generated by GENSCRN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

REGION 0
REGIONAL m.currarea, m.talkstat, m.compatstat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compatstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.currarea = SELECT()

```

Window definitions

```

IF NOT VEXIST("w_dview")
DEFINE WINDOW w_dview
FROM INT((SRON()-19)/2), INT((SCOL()-80)/2)
TO INT((SRON()-19)/2)+18, INT((SCOL()-80)/2)+79
TITLE "Disease Description"
FLOAT
NOCLOSE
SHADOW
DOUBLE
COLOR SCHEME 5
ENDIF

```

W8 Screen Layout

```

REGION 1
IF VISIBLE("w_dview")
ACTIVATE WINDOW w_dview SAME
ELSE
ACTIVATE WINDOW w_dview NOSHOW

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96

```

```

L-ENDIF
@ 1,0 EDIT mhelp ;
SIZE 15,78,0 ;
DEFAULT " " ;
SCROLL ;
MODIFY ;
@ 16,34 GET mbuttons ;
PICTURE "g=HT OK" ;
SIZE 1,8,4 ;
DEFAULT " " ;
-IF NOT VISIBLE("w_dview")
ACTIVATE WINDOW w_dview
ENDIF
READ CYCLE MODAL
RELEASE WINDOW w_dview
SELECT (m.currarea)
REGION 0
-IF m.talkstat = "ON"
SET TALK ON
ENDIF
-IF m.compatstat = "ON"
SET COMPATIBLE ON
ENDIF
* EOF: W8.ACT

```

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

07/02/92      W9.SPR      11:15:35

Author's Name
Copyright (c) 1992 Company Name
Address
City,      Zip
Description:
This program was automatically generated by GENSCREEN.

REGION 0
REGIONAL m.curraea, m.talkstat, m.compatat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FORPLUS

Window definitions

IF NOT EXIST("w.popsag")
DEFINE WINDOW w.popsag
FROM INT((SROW()-8)/2),INT((SCOL()-36)/2)
TO INT((SROW()-8)/2)+7,INT((SCOL()-36)/2)+35;
TITLE "Warning";
FLOAT;
NOCLOSE;
SHADOW;
DOUBLE;
COLOR SCHEME 7
ENDIF

W9 Screen Layout

REGION 1
IF WISIBLE("w.popsag")
ACTIVATE WINDOW w.popsag SAME
ELSE
ACTIVATE WINDOW w.popsag MOSHOW
ENDIF
@ 1,1 SAY errstr;
SIZE 3,31
@ 5,14 GET mbuttons;

```

```

67 PICTURE "RIGHT OK";
68 SIZE 1,4,1;
69 DEFAULT;
70
71 IF NOT WISIBLE("w.popsag")
72 ACTIVATE WINDOW w.popsag
73 ENDIF
74
75 READ CYCLE MODAL
76
77 RELEASE WINDOW w.popsag
78
79 REGION 0
80 IF m.talkstat = "ON"
81 SET TALK ON
82 ENDIF
83 IF m.compatat = "ON"
84 SET COMPATIBLE ON
85 ENDIF
86
87 *: EOF: W9.AC1

```

07/20/92	W10.SPR	10:00:36
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

```

```

REGION 0
REGIONAL m.curreee, m.talkstat, m.compatat

```

```

IF SET("TALK") = "ON"
  SET TALK Off
  m.talkstat = "ON"
ELSE
  m.talkstat = "Off"
ENDIF
m.compatat = SET("COMPATIBLE")
SET "COMPATIBLE FORPLUS"

```

Window definitions

```

IF NOT EXIST(" q710lgdb1")
  DEFINE WINDOW q710lgdb1
  FROM INT((SROW()-9)/2),INT((SCOL()-35)/2)
  TO INT((SROW()-9)/2)+8,INT((SCOL()-35)/2)+34
  TITLE "Confirm"
  FLOAT
  CLOSE
  SHADOW
  COLOR SCHEME 1
ENDIF

```

W10 Screen Layout

```

REGION 1
IF WISIBLE(" q710lgdb1")
  ACTIVATE WINDOW q710lgdb1 SAME
ELSE
  ACTIVATE WINDOW q710lgdb1 NOSHOWN
ENDIF
@ 1,1 SAY m.prompt
@ 4,31 SIZE 4,31
@ 6,7 GET m.buttons
PICTURE "OK;Cancel"

```

W10.AC1 10-14-92 10:51a

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
86

```

```

SIZE 1,8,1
DEFAULT {
  IF NOT WISIBLE(" q710lgdb1")
    ACTIVATE WINDOW q710lgdb1
  ENDIF
  READ CYCLE MODAL
  RELEASE WINDOW q710lgdb1
  #REGION 0
  IF m.talkstat = "ON"
    SET TALK ON
  ENDIF
  IF m.compatat = "ON"
    SET COMPATIBLE ON
  ENDIF
  *: EOF: W10.AC1

```



```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

07/20/92	W11.SPR	10:01:02
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

REGION 0
REGIONAL m.currares, m.talkstat, m.compatat

```

```

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT EXIST("w_pend")
  DEFINE WINDOW w_pend
  FROM INT((SROW()-19)/2), INT((SCOL()-59)/2) ;
  TO INT((SROW()-19)/2)+18, INT((SCOL()-59)/2)+58 ;
  TITLE "Pending Encounters" ;
  FLOAT ;
  NOCLOSE ;
  SHADOW ;
  DOUBLE ;
  COLOR SCHEME 5
ENDIF

```

W11 Setup Code - SECTION 2

```

REGION 1
mchoice = 1

```

W11 Screen Layout

```

67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *

```

```

REGION 1
IF WISIBLE("w_pend")
  ACTIVATE WINDOW w_pend SAME
ELSE
  ACTIVATE WINDOW w_pend NOSHOW
ENDIF
@ 6,1 GET mchoice ;
  PICTURE "a2m" ;
  FROM mlist ;
  SIZE 9,55 ;
  DEFAULT 1 ;
  COLOR SCHEME 6
@ 16,11 GET mbuttons ;
  PICTURE "a2m" Resume;New;\?Cancel" ;
  SIZE 1,8,5 ;
  DEFAULT 1 ;
@ 3,1 SAY "date"
@ 1,1 SAY "patient"
@ 1,10 SAY "key" ;
  SIZE 1,10
@ 3,10 SAY "update" ;
  SIZE 1,10

IF NOT WISIBLE("w_pend")
  ACTIVATE WINDOW w_pend
ENDIF
READ CYCLE MODAL
RELEASE WINDOW w_pend

REGION 0
IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compatat = "ON"
  SET COMPATIBLE ON
ENDIF
=: EOF: W11.ACT

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

10/13/92	W12.SPR	14:17:09
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

REGION 0
REGIONAL m.curreas, m.talkstat, m.compatat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "Off"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT WEXIST("w_encount")
DEFINE WINDOW w_encount ;
FROM 1, 0 ;
TO 22, 79 ;
TITLE "Patient Encounter" ;
NOFLOAT ;
NOCLOSE ;
DOUBLE ;
COLOR SCHEME 5
ENDIF
IF NOT WEXIST("w_warning")
DEFINE WINDOW w_warning ;
FROM 23, 0 ;
TO 24, 79 ;
NOFLOAT ;
NOCLOSE ;
NONE ;
COLOR SCHEME 5
ENDIF

```

W12 Screen Layout

W12.AC1 10-14-92 10:51a

```

67 #REGION 1
68 IF WVISIBLE("w_encount")
69 ACTIVATE WINDOW w_encount SAME
70 ELSE
71 ACTIVATE WINDOW w_encount NOSHOW
72 ENDIF
73 @ 0,5 GET me_dateenc ;
74 SIZE 1,8 ;
75 DEFAULT ( / / ) ;
76 DISABLE
77 @ 0,20 GET me_timeenc ;
78 SIZE 1,5 ;
79 DEFAULT " " ;
80 DISABLE
81 @ 0,34 GET me_state ;
82 SIZE 1,4 ;
83 DEFAULT " " ;
84 DISABLE
85 @ 0,52 GET me_providr ;
86 SIZE 1,24 ;
87 DEFAULT " " ;
88 DISABLE
89 @ 1,13 GET me_lastname ;
90 SIZE 1,28 ;
91 DEFAULT " " ;
92 DISABLE
93 @ 1,49 GET me_firstname ;
94 SIZE 1,22 ;
95 DEFAULT " " ;
96 PICTURE "a1" ;
97 DISABLE
98 @ 2,4 GET me_ssn ;
99 SIZE 1,11 ;
100 DEFAULT " " ;
101 PICTURE "99-99-9999" ;
102 DISABLE
103 @ 2,31 GET me_sex ;
104 SIZE 1,6 ;
105 DEFAULT " " ;
106 PICTURE "a1" ;
107 DISABLE
108 @ 2,47 GET me_age ;
109 SIZE 1,3 ;
110 DEFAULT " " ;
111 DISABLE
112 @ 4,0 GET mckcompl ;
113 PICTURE "g-HN \<Complaint" ;
114 SIZE 1,12,1 ;
115 DEFAULT 1 ;
116 VALID editcomp()
117 @ 4,13 GET me_area ;
118 SIZE 1,24 ;
119 DEFAULT " " ;
120 DISABLE
121 @ 5,0 TO 13,38
122 @ 5,15 GET mckhndp ;
123 PICTURE "g-HN \<Symptom" ;
124 SIZE 1,9,1 ;
125 DEFAULT 1 ;
126 VALID editthndp()
127 @ 6,1 GET mckcond ;
128 PICTURE "g-HN" ;
129 FROM me_cond ;
130 SIZE 7,57 ;
131 DEFAULT 1 ;
132 VALID editcond(mckcond) ;

```

Page 1 of 4

265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289

V12 Cleanup Code

```

REGION 1
-IF quitting
  RELEASE WINDOW w_encount
  IF MEXIST('w_warning')
    RELEASE WINDOW w_warning
  ENDIF
ENDIF
ENDIF

```

mtmax	VALID
Function Origin:	
from Screen:	W12,
Variable:	mtmax
Called By:	VALID
Snippet Number:	1
Record Number: 23	

```

FUNCTION qpvolumcrg 22 intmax VALID
#REGION 1
-IF NOT MONOP("w_treat")
DO w12c.spr
SHOW GETS
ENDIF

```

_09VOLUME24
 Function Origin:
 From Screen: W12, Record Number: 27
 Variable: mtrnorm
 Called By: mtrnorm
 Valid Clause: VALID Clause
 Snippet Number: 2

```
FUNCTION q9v0um2m
#REGION 1
=RELEASE("w treat")
22 mtnorm VALID
```

```

      _09VOLUME01
      Function Origin:
      From Screen:
      Variable:
      Called By:
      Snippet Number:
      msnorm VALID
      w12, Record Number: 36
      msnorm
      VALID Clause
      3

```

W12.AC1 10-14-92 10:51a

331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355

```
FUNCTION q9v0mddi
#REGION 1
=RELEASE("w handp")
END MSFORM VALID
```

```

_09VOLUMEJJ      memex  VALID
Function Origin:
From Screen:      W12,      Record Number: 37
Variable:         memex
Called By:        VALID
Subscript Number: 4

```

```

FUNCTION _q9v0umdjj      24  msmax valid
#REGION 1
IF NOT MONTOP("w_handp")
DO w12a.spr
SHOW GETS
ENDIF

```

```

memnorm __99VOUNDQF
Function Origin:
  From Screen:
  Variable:
  Called By:
  Snippet Number:
  memnorm VALID
  W12, Record Number: 38
  memnorm
  VALID Clause
  5

```

```

FUNCTION _q9v0undqf
#REGION 1
#RELEASE(MW assist™)

```

```

memmax VALID
Function Origin:
  From Screen:
  Variable:
  Called By:
  Snippet Number:
  W12, Record Number: 39
  memmax
  VALID Clause
  6

```

```

FUNCTION _qpvouncing
REGION 1
IF NOT MONITOR("w_assistn")
DO w12b.spr
SHOW GETS
ENDIF
ENDIF
22 memmax VALID

```

```
397 *
398 *
399 *
400 *
401 *
402 *
403 *
404 *
405 *
406 *
407 *
408 *
409 *
410 *
411 *
412 *
413 *
414 *
416 *
```

W12.AC1 10-14-92 10:51a
Read Level Valid

_Q9VOLUME80

Function Origin:
From Screen: Multiple Screens
Called By: READ Statement
Snippet Number: 7

```
FUNCTION _q9vome80    && Read Level Valid
* Valid Code from screen: W12
#REGION
  IF NOTOP("W12e")
    ACTIVATE WINDOW w_encount
  ENDIF
*: EOF: W12.AC1
```

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

07/20/92	W13.SPR	10:01:36
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

#REGION 0
REGIONAL m.curarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT EXIST("w_undo")
DEFINE WINDOW w_undo
FROM INT((SROW()-12)/2), INT((SCOL()-32)/2)
TO INT((SROW()-12)/2)+1, INT((SCOL()-32)/2)+31
TITLE "Undo Options"
FLOAT
NOCLOSE
SHADOW
DOUBLE
COLOR SCHEME 1
ENDIF

```

W13 Screen Layout

```

#REGION 1
IF WISIBLE("w_undo")
ACTIVATE WINDOW w_undo SAME
ELSE
ACTIVATE WINDOW w_undo NOSHOW
ENDIF
@ 1,2 GET mchoice
PICTURE "alt"
FROM mped

```

```

67 SIZE 7,26 ;
68 DEFAULT 1 ;
69 COLOR SCHEME 2
70 @ 9,5 GET mbuttons ;
71 PICTURE "PRINT OK,\Cancel" ;
72 SIZE 1,8,1 ;
73 DEFAULT 1 ;
74
75 IF NOT WISIBLE("w_undo")
76 ACTIVATE WINDOW w_undo
77 ENDIF
78 READ CYCLE MODAL
79 RELEASE WINDOW w_undo
80
81 #REGION 0
82 IF m.talkstat = "ON"
83 SET TALK ON
84 ENDIF
85 IF m.compstat = "ON"
86 SET COMPATIBLE ON
87 ENDIF
88 =: EOF: W13.AC1
89
90
91

```

07/20/92	W14.SPR	10:00:31
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRM.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

```

```

REGION 0
REGIONAL m.curree, m.talkstat, m.compatet

```

```

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compatet = SET("COMPATIBLE",
SET COMPATIBLE FORPLUS

```

Window definitions

```

31
32
33
34
35
36
37

```

```

IF NOT EXIST("w_list")
  DEFINE WINDOW w_list
  FROM INT((SROW()-14)/2), INT((SCOL()-52)/2)
  TO INT((SROW()-14)/2)+15, INT((SCOL()-52)/2)+51
  TITLE "Change Data"
  FLOAT
  NOCLOSE
  SHADOW
  DOUBLE
  COLOR SCHEME 1
ENDIF

```

W14 Screen Layout

```

51
52
53
54
55
56
57

```

```

REGION 1
IF VISIBLE("w_list")
  ACTIVATE WINDOW w_list SAME
ELSE
  ACTIVATE WINDOW w_list NOSHOW
ENDIF
@ 1,0 GET mchoice
  PICTURE "dat"
  FROM mped

```

```

58
59
60
61
62
63
64
65
66

```

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91

```

```

SIZE 9,50 ;
DEFAULT 1 ;
COLOR SCHEME 2
@ 1,1 GET mbuttons
  PICTURE "m=NT OK;7Cancel" ;
  SIZE 1,8,1 ;
  DEFAULT 1 ;
  IF NOT VISIBLE("w_list")
    ACTIVATE WINDOW w_list
  ENDIF
  READ CYCLE MODAL
  RELEASE WINDOW w_list
  REGION 0
  IF m.talkstat = "ON"
    SET TALK ON
  ENDIF
  IF m.compatet = "ON"
    SET COMPATIBLE ON
  ENDIF
  ": EOF: W14.AC1

```

67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 21

0700Y1CA9	mchoice	VALID
Function Origin:		
From Screen:	WP3	Record Number: 2
Variable:	mchoice	
Called By:	VALID Clause	
Object Type:	List	
Snippet Number:	1	

07/20/92	MP4.SPR	00:01:46
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

REGION 0
REGIONAL m.curarea, m.talkstat, m.compstat

```

```

IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FORPLUS

```

Window definitions

```

IF NOT EXIST("mp4")
DEFINE WINDOW mp4
FROM INT((SRON()-13)/2), INT((SCOL()-56)/2)
TO INT((SRON()-13)/2)+12, INT((SCOL()-56)/2)+55
TITLE "Encounter to Report"
NOFLOAT
NOCLOSE
SHADOW
DOUBLE
COLOR SCHEME 5
ENDIF

```

MP4 Screen Layout

```

REGION 1
IF VISIBLE("mp4")
ACTIVATE WINDOW mp4 SAME
ELSE
ACTIVATE WINDOW mp4 NOSHOW
ENDIF
@ 1,20 GET msn ;
SIZE 1,12 ;
DEFAULT " " ;

```

```

67 PICTURE "MR 999-99-9999"
68 @ 1,3 SAY "Social Security:"
69 @ 3,3 SAY "Last Name:"
70 @ 5,3 SAY "First Name:"
71 @ 3,20 GET mlast ;
72 SIZE 1,31 ;
73 DEFAULT " " ;
74 @ 5,20 GET mfirst ;
75 SIZE 1,31 ;
76 DEFAULT " " ;
77 @ 7,3 SAY "Encounter date:"
78 @ 7,20 GET mdateenc ;
79 SIZE 1,10 ;
80 DEFAULT " / / " ;
81 @ 9,11 GET mbuttons ;
82 PICTURE "PRINT \\\<OK>\<Browse>\<Cancel>" ;
83 SIZE 1,8,3 ;
84 DEFAULT " " ;
85
86 IF NOT VISIBLE("mp4")
87 ACTIVATE WINDOW mp4
88 ENDIF
89
90 READ CYCLE MODAL
91
92 RELEASE WINDOW mp4
93
94 #REGION 0
95 IF m.talkstat = "ON"
96 SET TALK ON
97 ENDIF
98 IF m.compstat = "ON"
99 SET COMPATIBLE ON
100 ENDIF
101
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 #REGION 1
110 RETURN mbuttons
111 *: EOF: MP4.AC1
112
113

```

MP4 Cleanup Code

07/20/92	WP5.SPR	10:01:51
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

```
#REGION 0
REGIONAL m.curraea, m.talkstat, m.compatat
```

```
IF SET("TALK") = "ON"
SET TALK OFF
ELSE
m.talkstat = "ON"
m.talkstat = "OFF"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
```

Window definitions

```
IF NOT MEXIST("wp4")
DEFINE WINDOW wp4
FROM INT((SROW()-17)/2), INT((SCOL()-59)/2)
TO INT((SROW()-17)/2)+16, INT((SCOL()-59)/2)+58 ;
NOFLOAT ;
NOCLOSE ;
SHADOW ;
DOUBLE ;
COLOR SCHEME 5
ENDIF
```

MP5 Setup Code - SECTION 2

```
#REGION 1
mchoice = 1
```

MP5 Screen Layout

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```
67 #REGION 1
68 IF WISIBLE("wp4")
69 ACTIVATE WINDOW wp4 SAME
70 ELSE
71 ACTIVATE WINDOW wp4 NOSHOW
72 ENDIF
73 @ 4,0 GET mchoice ;
74 FROM alist ;
75 SIZE 9,55 ;
76 DEFAULT 1 ;
77 COLOR SCHEME 6
78 @ 14,18 GET mbuttons ;
79 PICTURE "gHT \OK;\?Cancel" ;
80 SIZE 18,5 ;
81 DEFAULT 1 ;
82 @ 2,0 SAY "Date"
83 @ 0,0 SAY "Patient"
84 @ 0,9 SAY mkey ;
85 SIZE 1,45 ;
86 @ 2,9 SAY mdate ;
87 SIZE 1,10 ;
88
89
90 IF NOT WISIBLE("wp4")
91 ACTIVATE WINDOW wp4
92 ENDIF
93 READ CYCLE MODAL
94 RELEASE WINDOW wp4
95
96 #REGION 0
97 IF m.talkstat = "ON"
98 SET TALK ON
99 ENDIF
100 IF m.compatat = "ON"
101 SET COMPATIBLE ON
102 ENDIF
103
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 #REGION 1
115 RETURN mchoice
116 *: EOF: WP5.AC1
117
```

MP5 Cleanup Code

07/20/92	MP6.SPR	10:01:55
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *

```

```

18 #REGION 0
19 REGIONAL m.currares, m.talkstat, m.compatat
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

```

22 IF SET("TALK") = "ON"
23   SET TALK OFF
24   m.talkstat = "ON"
25 ELSE
26   m.talkstat = "OFF"
27 ENDIF
28 m.comps.at = SET("COMPATIBLE")
29 SET COMPATIBLE FOXPLUS
30
31
32
33
34
35
36
37

```

Window definitions

```

38 IF NOT WEXIST("wp6")
39   DEFINE WINDOW wp6 ;
40   FROM INT((SROW()-20)/2), INT((SCOL()-77)/2) ;
41   TO INT((SROW()-20)/2)+19, INT((SCOL()-77)/2)+76 ;
42   TITLE "Report Editing" ;
43   NOFLOAT ;
44   NOCLOSE ;
45   SHADOW ;
46   DOUBLE ;
47   COLOR SCHEME 5
48 ENDIF
49
50
51
52
53
54
55
56
57
58

```

MP6 Screen Layout

```

59 #REGION 1
60 IF WISIBLE("wp6")
61   ACTIVATE WINDOW wp6 SAME
62 ELSE
63   ACTIVATE WINDOW wp6 HOSHOW
64 ENDIF
65 @ 1,1 EDIT mstr ;
66 SIZE 15,73,0 ;
67 DEFAULT ** ;

```

```

67 TAB ;
68 SCROLL
69 @ 17,23 GET mbuttons ;
70 PICTURE "g=HT \<Print;\<Save;\<Cancel" ;
71 SIZE 1,8,4 ;
72 DEFAULT 1 ;
73
74 IF NOT WISIBLE("wp6")
75   ACTIVATE WINDOW wp6
76 ENDIF
77
78 READ CYCLE MODAL
79
80 RELEASE WINDOW wp6
81
82 #REGION 0
83 IF m.talkstat = "ON"
84   SET TALK ON
85 ENDIF
86 IF m.compatat = "ON"
87   SET COMPATIBLE ON
88 ENDIF
89 *: EOF: MP6.AC1
90

```

07/20/92	WP7.SPR	10:02:00
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

REGION 0
REGIONAL m.curarea, m.talkstat, m.compatat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT MEXIST("wp7")
DEFINE WINDOW wp7 ;
FROM INT((ROW()-13)/2), INT((COL()-43)/2) ;
TO INT((ROW()-13)/2)+12, INT((COL()-43)/2)+42 ;
TITLE "Print Option" ;
FLOAT ;
NOCLOSE ;
SHADOW ;
DOUBLE ;
COLOR SCHEME 5
ENDIF

```

WP7 Setup Code - SECTION 2

```

REGION 1
PRIVATE mchoice, mprintfile, mfilename, i
mchoice = 1
mfilename = "sf600.txt"
mprintfile = mfilename

```

WP7 Screen Layout

```

67 *
68 *
69 *
70 *
71 #REGION 1
72 IF WISIBLE("wp7")
73 ACTIVATE WINDOW wp7 SAME
74 ELSE
75 ACTIVATE WINDOW wp7 NOSHOW
76 ENDIF
77 @ 0,0 TO 3,40
78 @ 5,0 TO 8,40
79 @ 6,1 SAY "Filename:"
80 @ 6,11 GET mprintfile ;
81 SIZE 1,29 ;
82 DEFAULT " " ;
83 VALID q7101177m()
84 @ 10,9 GET mbuttons ;
85 PICTURE "p*HT OK;\?Cancel" ;
86 SIZE 1,8,5 ;
87 DEFAULT " " ;
88 VALID q7101177f()
89 @ 1,24 GET mchoice ;
90 PICTURE "p*RVN File;Printer" ;
91 SIZE 1,11,0 ;
92 DEFAULT " " ;
93 VALID q7101177oj()
94 @ 1,1 SAY "Print Encounter to:"
95
96 IF NOT WISIBLE("wp7")
97 ACTIVATE WINDOW wp7
98 ENDIF
99
100 READ CYCLE MODAL
101
102 RELEASE WINDOW wp7
103
104 #REGION 0
105 IF m.talkstat = "ON"
106 SET TALK ON
107 ENDIF
108 IF m.compatat = "ON"
109 SET COMPATIBLE ON
110 ENDIF
111
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 FUNCTION q7101177m @L mprintfile VALID
128 #REGION 1
129 mfilename = mprintfile
130 RETURN .T.
131
132 *

```

_q7101177m	mprintfile VALID
Function Origin:	
From Screen:	wp7
Variable:	mprintfile
Called By:	VALID Clause
Object Type:	Field
Snippet Number:	1
Record Number: 5	

```

133 *
134 *
135 *
136 *
137 *
138 *
139 *
140 *
141 *
142 *
143 *
144 *
145 *
146 *
147 *
148 *
149 *
150 *
151 *
152 *
153 *
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *

```

_q7101177 mbuttons VALID

Function Origin:

From Screen: WP7, Record Number: 6

Variable: mbuttons

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 2

```

145 FUNCTION _q7101177      22 mbuttons VALID
146 #REGION 1
147 IF mbuttons = 1
148 IF EMPTY(mprintfile)
149 DO sprint WITH mprintfile, mtext
150 ENDIF
151 ENDIF
152 RETURN
153
154 *
155 *
156 *
157 *
158 *
159 *
160 *
161 *
162 *
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *

```

_q7101170j mchoice VALID

Function Origin:

From Screen: WP7, Record Number: 7

Variable: mchoice

Called By: VALID Clause

Object Type: Radio Button

Snippet Number: 3

```

168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *
176 *
177 *
178 *
179 *
180 *

```

_q7101170j 22 mchoice VALID

#REGION 1

IF mchoice = 1

mprintfile = mfilename

CURSOR = OBJNUM(mprintfile)

ELSE

mprintfile = "PRINT"

ENDIF

SHOW GETS

RETURN .T.

*: EOF: MP7.AC1

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

07/20/92	WP8.SPR	10:02:06
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCN.		

```

REGION 0
REGIONAL m.curraea, m.talkstat, m.compatat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compatat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT MEXIST("wp3")
DEFINE WINDOW wp3
FROM INT((SHOW()-16)/2), INT((SCOL()-41)/2) ;
TO INT((SHOW()-16)/2)+15, INT((SCOL()-41)/2)+40 ;
TITLE "Patient In File" ;
NOFLOAT ;
NOCLOSE ;
SHADOW ;
DOUBLE ;
COLOR SCHEME 5
ENDIF

```

WP8 Setup Code - SECTION 2

```

REGION 1
mchoice=1

```

WP8 Screen Layout

```

67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

```

REGION 1
IF WISIBLE("wp3")
ACTIVATE WINDOW wp3 SAME
ELSE
ACTIVATE WINDOW wp3 NOSHOW
ENDIF
@ 0,2 GET mchoice ;
PICTURE "OK" ;
FROM mlst ;
SIZE 12,34 ;
DEFAULT 1 ;
COLOR SCHEME 6
@ 12,8 GET mbuttons ;
PICTURE "OK" ;
SIZE 1,8,5 ;
DEFAULT 1 ;
IF NOT WISIBLE("wp3")
ACTIVATE WINDOW wp3
ENDIF
READ CYCLE MODAL
RELEASE WINDOW wp3
REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compatat = "ON"
SET COMPATIBLE ON
ENDIF
* EOF: WP8.AC1

```

07/06/92	W12A.SPR	10:46:49
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCREEN.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

REGION 0
REGIONAL m.curarea, m.talkstat, m.compatstat
-IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
-ELSE
  m.talkstat = "OFF"
-ENDIF
m.compatstat = SET("COMPATIBLE")
SET COMPATIBLE FORPLUS

```

Window definitions

```

-IF NOT EXIST("w_handp")
  DEFINE WINDOW w_handp
  FROM INT((SROW()-21)/2), INT((SCOL()-80)/2) ;
  TO INT((SROW()-21)/2)+20, INT((SCOL()-80)/2)+79 ;
  NOFLOAT ;
  NOCLOSE ;
  NONE ;
  COLOR SCHEME 5
-ENDIF

```

W12A Screen Layout

```

REGION 1
-IF VISIBLE("w_handp")
  ACTIVATE WINDOW w_handp SAME
-ELSE
  ACTIVATE WINDOW w_handp NOSHOW
-ENDIF
@ 0,0 TO 20,79
@ 1,2 GET second ;
  PICTURE "9999" ;
FROM me cond ;
SIZE 19,76 ;

```

```

67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *
95 *
96 *
97 *
98 *
99 *
100 *
101 *
102 *
103 *
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *

```

```

DEFAULT 1 ;
VALID editcond(mcond1) ;
COLOR SCHEME 6
@ 0,36 GET mchandp ;
  PICTURE "9999" (<Symptom" ;
  SIZE 1,9,1 ;
  DEFAULT 1 ;
  VALID editcond(mcond1) ;
  PICTURE "9999" ;
  SIZE 1,1,1 ;
  DEFAULT 0 ;
  VALID q740n3u9y(
    @ 0,76 SAY CHR(31) ;
    SIZE 1,1 ;
    @ 0,77 GET memax ;
    PICTURE "9999" ;
    SIZE 1,1,1 ;
    DEFAULT 0 ;
    VALID q740n3u9y(
      @ 0,77 SAY CHR(30) ;
      SIZE 1,1 ;

```

```

-IF NOT VISIBLE("w_handp")
  ACTIVATE WINDOW w_handp
-ENDIF

```

```

READ CYCLE MODAL

```

```

RELEASE WINDOW w_handp

```

```

REGION 0
-IF m.talkstat = "ON"
  SET TALK ON
-ENDIF
-IF m.compatstat = "ON"
  SET COMPATIBLE ON
-ENDIF

```

```

_q740n3u9y      memax VALID
Function Origin:
From Screen:    W12A,      Record Number:  5
Variable:       memax
Called By:      VALID Clause
Snippet Number:  1

```

```

FUNCTION _q740n3u9y      24 memax VALID
REGION 1
-RELEASE("w_handp")

```

```

_q740n3u9y      memax VALID
Function Origin:
From Screen:    W12A,      Record Number:  7
Variable:       memax
Called By:      VALID Clause

```

Snippet Number: 2

133 *
 134 *
 135 *
 136 *
 137 FUNCTION q74On3Jhf 24 memax VALID
 138 REGION 1
 139 IF NOT MONTOP("W12A")
 140 DO W12a.spr
 141 ENDIF
 142 *:
 143 EOF: W12A.AC1


```

67 VALID _q740n3xwa( )
68 @ 1,1 GET mdiagl ;
69 PICTURE "Q84M" ;
70 FROM mdiagl ;
71 SIZE 19,77 ;
72 DEFAULT 1 ;
73 VALID editdiag(mdiagl) ;
74 COLOR SCHEME 6
75 @ 0,38 GET mckassiat ;
76 PICTURE "Q84M" \<Assiat" ;
77 SIZE 1,8,1 ;
78 DEFAULT 1 ;
79 VALID editassiat( )
80 @ 0,78 GET memax ;
81 PICTURE "Q84M" " ;
82 SIZE 1,1,1 ;
83 DEFAULT 0 ;
84 VALID _q740n3y5o( )
85 @ 0,78 SAY CHR(30) ;
86 SIZE 1,1
87 @ 0,77 SAY CHR(31) ;
88 SIZE 1,1
89
90 [ IF NOT WVISIBLE("W_assist")
91     ACTIVATE WINDOW W_assist
92 ]
93 ENDIF
94
95 READ CYCLE MODAL
96
97 RELEASE WINDOW W_assist
98
99 #REGION 0
100 [ IF W_talkstat = "ON"
101     SET TALK ON
102 ]
103 ENDIF
104 [ IF M_comstat = "ON"
105     SET COMPATIBLE ON
106 ]
107 ENDIF
108
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 FUNCTION _q740n3xwa
121 #REGION 1
122 =RELEASE("W_assist")
123
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *

```

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

07/06/92	V128.SPR	10:46:54
----------	----------	----------

```

Author's Name
Copyright (c) 1992 Company Name
Address
City, Zip
Description:
This program was automatically generated by GENSCRN.

```

```

#REGION 0
REGIONAL m.curraea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT MEXIST("w_assist")
DEFINE WINDOW w_assist ;
FROM INT((SROW()-21)/2), INT((SCOL()-80)/2) ;
TO INT((SROW()-21)/2)+20, INT((SCOL()-80)/2)+79 ;
NOFLOAT ;
NOCLOSE ;
NONE ;
COLOR SCHEME 5
ENDIF

```

W128 Screen Layout

```

#REGION 1
IF WVISIBLE("w_assist")
ACTIVATE WINDOW w_assist SAME
ELSE
ACTIVATE WINDOW w_assist NOSHOW
ENDIF
@ 0,0 TO 20,79
@ 0,77 GET memorm ;
PICTURE "g*lvw";
SIZE 1,1 ;
DEFAULT 0 ;

```

Called By: VALID Clause
Snippet Number: 2

133
134
135
136
137
138
139
140
141
142
144

```

*
*
*
*
FUNCTION q740n3y5o 24 memax VALID
#REGION 1
IF NOT MONTOP("w_assist")
DO w12b.spr
ENDIF
*: EOF: W12B.AC1
    
```

07/06/92	W12C.SPR	10:46:59
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRM.		

```

1 *
2 *
3 *
4 *
5 *
6 *
7 *
8 *
9 *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

REGION 0
REGIONAL m.curarea, m.talkstat, m.compstat

```

```

IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

Window definitions

```

IF NOT EXISTS("w_treat")
DEFINE WINDOW w_treat
FROM INT((SROW()-21)/2), INT((SCOL()-80)/2) ;
TO INT((SROW()-21)/2)+20, INT((SCOL()-80)/2)+79 ;
TITLE "Treatment" ;
NOFLOAT ;
NOCLOSE ;
NONE ;
COLOR SCHEME 5
ENDIF

```

W12C Screen Layout

```

REGION 1
IF VISIBLE("w_treat")
ACTIVATE WINDOW w_treat SAME
ELSE
ACTIVATE WINDOW w_treat NOSHOW
ENDIF
@ 0,0 TO 20,79
@ 2,2 EDIT me_treat ;
SIZE 17,76,0 ;
DEFAULT " " ;

```

W12C.AC1 10-14-92 10:52a

```

67 SCROLL
68 @ 0,76 GET mtmax ;
69 PICTURE "999999" ;
70 SIZE 1,1,1 ;
71 DEFAULT 0 ;
72 VALID _q740n41w1( ;
73 @ 0,75 GET mtnorm ;
74 PICTURE "999999" ;
75 SIZE 1,1,1 ;
76 DEFAULT 0 ;
77 VALID _q740n423h( ;
78 @ 0,76 SAY CHR(31) ;
79 SIZE 1,1 ;
80 @ 0,75 SAY CHR(31) ;
81 SIZE 1,1 ;
82 @ 0,36 GET mcktreast ;
83 PICTURE "999999" ;
84 SIZE 1,12,1 ;
85 DEFAULT 1 ;
86 VALID _edit_treat
87
88 IF NOT VISIBLE("w_treat")
89 ACTIVATE WINDOW w_treat
90 ENDIF
91
92 READ CYCLE MODAL
93
94 RELEASE WINDOW w_treat
95
96 #REGION 0
97 IF m.talkstat = "ON"
98 SET TALK ON
99 ENDIF
100 IF m.compstat = "ON"
101 SET COMPATIBLE ON
102 ENDIF
103
104 *
105 *
106 *
107 *
108 *
109 *
110 *
111 *
112 *
113 *
114 *
115 *
116 *
117 *
118 *
119 *
120 *
121 *
122 *
123 *
124 *
125 *
126 *
127 *
128 *
129 *
130 *
131 *
132 *

```

```

IF NOT VISIBLE("w_treat")
ACTIVATE WINDOW w_treat
ENDIF

```

READ CYCLE MODAL

RELEASE WINDOW w_treat

#REGION 0

```
IF m.talkstat = "ON"
```

```
SET TALK ON
```

```
ENDIF
```

```
IF m.compstat = "ON"
```

```
SET COMPATIBLE ON
```

```
ENDIF
```

_q740n41w1	mtmax VALID
Function Origin:	W12C, Record Number: 4
From Screen:	mtmax
Variable:	VALID Clause
Called By:	1
Snippet Number:	

FUNCTION _q740n41w1 22 mtmax VALID

#REGION 1

```
IF NOT VISIBL("w_treat")
```

```
DO W12c.spr
```

```
ENDIF
```

_q740n423h	mtnorm VALID
Function Origin:	W12C, Record Number: 5
From Screen:	mtnorm
Variable:	

Page 1 of 2

Called By: VALID Clause
Snippet Number: 2

133
134
135
136
137
138
139
140
142

*
*
*
*
*
FUNCTION q740r423h 22 mtnorm VALID
#REGION 1
=RELEASE("w treat")
*: EOF: M12C.AC1

08/20/92	WP1.PRG	16:00:07
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRM.		

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

REGION 0
REGIONAL m.currarea, m.talkstat, m.compatstat

```

```

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compatstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.currarea = SELECT()

```

Window definitions

```

IF NOT EXISTS("wp1")
  DEFINE WINDOW wp1 ;
  FROM INT((SROW()-11)/2), INT((SCOL()-57)/2) ;
  TO INT((SROW()-11)/2)+10, INT((SCOL()-57)/2)+56 ;
  TITLE "Patient Selection" ;
  NOFLOAT ;
  NOCLOSE ;
  SHADOW ;
  DOUBLE ;
  COLOR SCHEME 5
ENDIF

```

WP1 Setup Code - SECTION 2

```

REGION 1
msecsec = " "
mlastname = ""
mfirstname = ""
mresult = ""

```

WP1 Screen Layout

```

REGION 1
IF NOT EXISTS("wp1")
  ACTIVATE WINDOW wp1 SAME
ELSE
  ACTIVATE WINDOW wp1 NOSHOW
ENDIF
0 1,3 SAY "Social Security: "
0 3,3 SAY "Lastname:"
0 5,3 SAY "Firstname:"
0 1,20 GET msecsec ;
  SIZE 1,11 ;
  DEFAULT " "
0 3,20 GET mlastname ;
  SIZE 1,31 ;
  DEFAULT " "
  PICTURE "911"
0 5,20 GET mfirstname ;
  SIZE 1,31 ;
  DEFAULT " "
  PICTURE "911"
0 8,9 GET mchoice ;
  PICTURE "9999" (\COL;\CROUSE;\CANCEL" ;
  SIZE 1,10,3 ;
  DEFAULT " "
  VALID _qdbdyardut()
IF NOT EXISTS("wp1")
  ACTIVATE WINDOW wp1
ENDIF
READ CYCLE MODAL
RELEASE WINDOW wp1
SELECT (m.currarea)
REGION 0
IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compatstat = "ON"
  SET COMPATIBLE ON
ENDIF

```

WP1 Cleanup Code

```

REGION 1
DEACTIVATE WINDOW wp1
RELEASE WINDOW wp1
RETURN mresult

```

choice VAL10

From Screen: WP1, Record Number: 8
Variable: mchoice
Called By: VALID Ctlause
Object Type: Push Button
Snippet Number: 1

```

FUNCTION _qbd0yrbu      22  mchoice VALID
PREGION 1
DO CASE
CASE mchoice = 1
    mresult = search(mocsec, alastname, mfirstname)
    IF EMPTY(mresult)
        CLEAR READ
    ENDIF
CASE mchoice = 2
    mresult = browsepat()
    IF EMPTY(mresult)
        CLEAR READ
    ENDIF
CASE mchoice = 3
    mresult = ""
    CLEAR READ
ENDCASE
RETURN mresult
* EOF: WP1.ACT

```

MP2.ACT 10-14-92 10:52a

Page 1 of 3

```

133 DEFAULT " " ;
134 PICTURE "QA1" ;
135 @ 8,19 GET mdo ;
136 SIZE 1,8 ;
137 DEFAULT ( / / ) ;
138 VALID _q8d0twktt( ) ;
139 @ 10,19 GET msk ;
140 SIZE 1,1 ;
141 DEFAULT " " ;
142 PICTURE "QA1" ;
143 @ 12,19 GET mrank ;
144 SIZE 1,28 ;
145 DEFAULT " " ;
146 PICTURE "QA1" ;
147 @ 14,13 GET maction ;
148 PICTURE "g" "H" \OK;\<CANCEL" ;
149 SIZE 1,8,3 ;
150 DEFAULT 1 ;
151 VALID _q8d0twdv1( ) ;
152
153 IF NOT UNVISIBLE("wp2")
154   ACTIVATE WINDOW wp2
155 ENDIF
156
157 READ CYCLE MODAL
158
159 RELEASE WINDOW wp2
160 SELECT (m.currarea)
161
162
163 #REGION 0
164 IF m.talkstat = "CON"
165   SET TALK ON
166 ENDIF
167 IF m.comstat = "CON"
168   SET COMPATIBLE ON
169 ENDIF
170
171
172
173
174
175
176
177
178
179 #REGION 1
180 =pophelp( )
181 DEACTIVATE WINDOW wp2
182 RELEASE WINDOW wp2
183 RETURN mfinal
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198

```

WP2 Cleanup Code

```

199
200 FUNCTION _q8d0twc24 24 mskn VALID
201 #REGION 1
202 IF mdoasn = mskn
203   invalid = .T.
204 ELSE
205   msel = SELECT( )
206   SELECT patient
207   SET ORDER TO 1
208   IF SEEK(mskn)
209     invalid = .F.
210     mskn = mdoasn
211     =popmessage("Duplicate Social Security Number")
212     CURSOR = CURSOR(mskn)
213     SHOW GETS
214   ELSE
215     invalid = .T.
216     mdoasn = mskn
217   ENDIF
218   SELECT (msel)
219   ENDIF
220   RETURN
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237 FUNCTION _q8d0twdkt 24 mdo VALID
238 #REGION 1
239 IF mdo > DATE( )
240   RETURN .F.
241 ELSE
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257 FUNCTION _q8d0twdv1 24 maction VALID
258 #REGION 1
259 DO CASE
260 CASE maction = 1
261   mfinal = .F.
262   mskn = STATRAM(mskn, " ")
263   DO CASE
264

```

Function Origin: mdo VALID

From Screen: WP2, Record Number: 13

Variable: mdo

Called By: VALID Clause

Object Type: Field

Snippet Number: 2

Function Origin: maction VALID

From Screen: WP2, Record Number: 16

Variable: maction

Called By: VALID Clause

Object Type: Push Button

Snippet Number: 3

Function Origin: mskn VALID

From Screen: WP2, Record Number: 9

Variable: mskn

Called By: VALID Clause

Object Type: Field

Snippet Number: 1


```

265 -CASE LEN(ALLTRIM(mtsen)) != 9
266   = popmessage("Invalid Social Security ID")
267   CURJOB = OBJNUM(msen)
268   merr = .t.
269   * CASE empty(mdob)
270   *   && hl106 no require field
271   *   = popmessage("Please enter patient date of birth")
272   *   CURJOB = OBJNUM(mdob)
273   *   merr = .t.
274   * CASE empty(msex)
275   *   = popmessage("Please enter patient sex")
276   *   CURJOB = OBJNUM(msex)
277   *   merr = .t.
278 -ENDCASE
279 -IF merr
280   SHOW GETS
281 -ELSE
282   mfinal = combined(msen,mlast,mfirst,mmiddle,DTOC(mdob),msex,mrank
283   => .STR(mpatid),STR(mrec))
284   mfinal = savrec(mnew, mfinal)
285   CLEAR READ
286 -ENDIF
287 -CASE maction = 2
288   mfinal = ""
289   CLEAR READ
290 -ENDCASE
291 RETURN mfinal
292 *: EOF: WP2.ACT

```

07/20/92	WP9.SPR	10:02:11
Author's Name Copyright (c) 1992 Company Name Address City, Zip Description: This program was automatically generated by GENSCRN.		

WP9 Setup Code - SECTION 1

```

1  *
2  *
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10 *
11 *
12 *
13 *
14 *
15 *
16 *
17 *
18 *
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *
32 *
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50 *
51 *
52 *
53 *
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66 *

```

```

#REGION 1
PARAMETER msyswar
IF PARAMETER() = 0
  RETURN
ENDIF
mtext = msyswar

#REGION 0
REGIONAL m.curarea, m.talkstat, m.compatstat

IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compatstat = SET("COMPATIBLE")
SET COMPATIBLE FORPLUS

```

Window definitions

```

IF NOT WEXIST("wp9")
  DEFINE WINDOW wp9 ;
  FROM 1, 0 ;
  TO 6, 76 ;
  NOFLOAT ;
  NOCLOSE ;
  COLOR SCHEME 7
ENDIF

```

WP9 Screen Layout

```

67 *
68 *
69 *
70 *
71 *
72 *
73 *
74 *
75 *
76 *
77 *
78 *
79 *
80 *
81 *
82 *
83 *
84 *
85 *
86 *
87 *
88 *
89 *
90 *
91 *
92 *
93 *
94 *

```

```

#REGION 1
IF WVISIBLE("wp9")
  ACTIVATE WINDOW wp9 SAME
ELSE
  ACTIVATE WINDOW wp9 MOSHOW
ENDIF
@ 0,1 SAY mtext ;
  SIZE 4,73

IF NOT WVISIBLE("wp9")
  ACTIVATE WINDOW wp9
ENDIF

READ CYCLE MODAL

RELEASE WINDOW wp9

#REGION 0
IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compatstat = "ON"
  SET COMPATIBLE ON
ENDIF
*: EOF: WP9.AC1

```

[illegible]

```

133 IF m.compatat = "ON"
134 SET COMPATIBLE ON
135 ENDIF
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194

```

WPATS Cleanup Code

```

*****
* This function return the result delimited by " "
*****
FUNCTION result
  PARAMETER mitem
  PRIVATE mpos
  mnew = .F.
  mpos = ASCAN(mlist, mitem)
  IF mpos > 0
    mfile = db(mlist[mpos+1], u, 1)
    IF UPPER(mfile) $ "SANS"
      mnew = .T.
    ENDIF
    mtemp = mlist[mpos] + u + mlist[mpos+1]
    mdata = getpatdata(mtemp)
    IF mnew
      mdata = savrec(mnew, mdata)
    ENDIF
  ELSE
    mdata = ""
  ENDIF
  RETURN mdata
*****
* This function call edit window
*****
PROCEDURE patedit
  PARAMETER mitem
  PRIVATE mpos
  mpos = ASCAN(mlist, mitem)
  IF mpos > 0
    mfile = db(mlist[mpos+1], u, 1)
    IF UPPER(mfile) $ "SANS"
      mnew = .T.
    ELSE
      mnew = .F.
    ENDIF
    mtemp = mlist[mpos] + u + mlist[mpos+1]
    mtext = getpatdata(mtemp)
    DEACTIVATE WINDOW wpats
    RELEASE WINDOW wpats
    mdata = wp2(mnew, mtext)
  ELSE
    mdata = ""
  ENDIF

```

```

195 ENDIF
196 RETURN mdata
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258

```

_q710151b2 mchoose VALID
 Function Origin: WPATS, Record Number: 3
 From Screen: mchoose
 Variable: VALID Clause
 Called By: List
 Object Type: 1
 Snippet Number: 1

_q710151j5 maction VALID
 Function Origin: WPATS, Record Number: 4
 From Screen: maction
 Variable: VALID Clause
 Called By: Push Button
 Object Type: 2
 Snippet Number: 2

```

*****
* Start the READ on the list box
FUNCTION _q710151b2      mchoose VALID
  REGION 1
  _CURSORJ = OBJNUM(mchoose)
*****
* SELECT: return the select item
* EDIT: Let the user edit the text and return
* New: Enter the new patient
* CANCEL: QUIT
*****
FUNCTION _q710151j5      maction VALID
  REGION 1
  DO CASE
  CASE maction = 1
    mpat = result(mchoose)
    CLEAR READ
  CASE maction = 2
    mpat = patedit(mchoose)
    CLEAR READ
  CASE maction = 3

```

```

259      mpat = patnew()
260      CLEAR READ
261      CASE maction = 4
262      DO moresearch WITH mcriter, mlst
263      _JUROBJ = OBJNUM(mchoose)
264      mpat = ""
265      SHOW GETS
266      SHOW GET maction, 4 disabled
267      CASE maction = 5
268      mpat = ""
269      CLEAR READ
270      ENDCASE
271      RETURN mpat
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

```

_q710L5TXD	Read Level When
Function Origin:	
From Screen:	MPATS
Called By:	READ Statement
Snippet Number:	3

```

FUNCTION _q710L5txd  && Read Level When
*
* When Code from screen: MPATS
*
#REGION 1
DO CASE
CASE msearch = 0
SHOW GET maction enabled
CASE msearch = 1
SHOW GET maction, 4 disabled
CASE msearch = 2
SHOW GET maction, 1 disabled
CASE msearch = 4
SHOW GET maction, 4 disabled
ENDCASE
*: EOF: MPATS.ACT

```

```

1 *****
2 Procedure file: C:\CAMD\WORK\YESNO.PRG
3
4 System: COMPUTER ASSISTED MEDICAL DIAGNOSIS
5 Author: ROY DOBBINS and HOA LY
6 Copyright (c) 1992, NAVAL HEALTH RESEARCH CENTER
7 Last modified: 11/26/91 2:50
8
9 Set by: STREPORT (procedure in DEMO.PRG)
10 : _OPORVE7X() (function in BACKUP.SPR)
11 : _OPORVEV4() (function in BACKUP.SPR)
12 : _OPORVD7U5() (function in RESTORE.SPR)
13
14 *****
15 Documented 10/14/92 at 10:52 FoxDoc version 2.10
16 *****
17 *
18 * 11/26/91 YESNO.PRG 02:50:49 *
19 *
20 *****
21 *
22 * Adam Green
23 *
24 * Copyright (c) 1991 Adam Green Seminars
25 * One Fenwick Hall
26 * Boston, MA 02174
27 *
28 * Description:
29 * This program was automatically generated by GENSCRN.
30 *
31 *****
32 *
33 *
34 *****
35 *
36 * YESNO Setup Code - SECTION 1
37 *
38 *****
39 *
40 *
41 REGION 1
42 PARAMETERS MESSAGE, OK, CANCEL
43 PRIVATE ALL
44
45 *HLL, If message was empty print a plain window just only need respon
46
47 ** ee
48 IF PARAMETERS() = 0
49 * Default to plain prompts
50 m.message = ""
51 ENDIF
52
53 * Ok and Cancel are used for prompts in the push buttons
54 IF PARAMETERS() = 1
55 * Default to normal prompts
56 m.ok = "OK"
57 m.cancel = "Cancel"
58 ENDIF
59
60 * Truncate any message longer than 50 characters
61 IF LEN( m.message ) > 50
62 * The message is centered in an ASAY
63 m.message = SUBSTR( m.message, 1, 50 )
64 ENDIF
65
66 PUSH KEY CLEAR

```

```

66 REGION 0
67 REGIONAL m.currearea, m.talkstat, m.compatat
68
69 IF SET("TALK") = "ON"
70 SET TALK OFF
71 m.talkstat = "ON"
72 ELSE
73 m.talkstat = "OFF"
74 ENDIF
75 m.compatat = SET("COMPATIBLE")
76 SET COMPATIBLE FOXPLUS
77
78 *****
79 *
80 * Window definitions
81 *
82 *
83 *
84 *****
85
86 IF NOT EXIST("yesno")
87 DEFINE WINDOW yesno ;
88 FROM INT((SRON()-7)/2), INT((SCOL()-48)/2) ;
89 TO INT((SRON()-7)/2)+6, INT((SCOL()-48)/2)+17 ;
90 FLOAT ;
91 NOCLOSE ;
92 SHADOW ;
93 DOUBLE ;
94 COLOR SCHEME 5
95 ENDIF
96
97 *****
98 *
99 * YESNO Screen Layout
100 *
101 *
102 *
103 *****
104
105 REGION 1
106 IF ACTIVABLE("yesno")
107 ACTIVATE WINDOW yesno SAME
108 ELSE
109 ACTIVATE WINDOW yesno NOSHOW
110 ENDIF
111 @ 1,2 SAY PADCC( m.message, 40 ) ;
112 @ SIZE 1,41
113 @ 3,8 GET m.answer ;
114 PICTURE "PHT \\\<OK;\<Cancel" ;
115 SIZE 1,12,5 ;
116 DEFAULT 1
117
118 IF NOT WVISIBLE("yesno")
119 ACTIVATE WINDOW yesno
120 ENDIF
121
122 READ CYCLE MODAL
123
124 RELEASE WINDOW yesno
125
126 REGION 0
127 IF m.talkstat = "ON"
128 SET TALK ON
129 ENDIF
130 IF m.compatat = "NON"
131 SET COMPATIBLE ON

```

```

132  ENDIF
133
134  *
135  * *****
136  * YESNO Cleanup Code *****
137  *
138  *
139  * *****
140  *
141  #REGION 1
142  POP KEY
143
144  * Convert the numeric value of m.answer from 1|2 to .T.|.F.
145  * If the user selected OK and didn't exit with Escape
146  IF m.answer = 1 AND LASTKEY() <> 27
147     RETURN .T.
148  ELSE
149     * Cancel or Escape returns false
150     RETURN .F.
151  ENDIF
152  * EOF: YESNO.ACT
153
154

```

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE October 1992		3. REPORT TYPE AND DATE COVERED Final Sept 91- Oct 92
4. TITLE AND SUBTITLE Computer Assisted Medical Diagnosis System Technical Manual			5. FUNDING NUMBERS Program Element: 63706N Work Unit Number: M0095.005-6103	
6. AUTHOR(S) Ly, Hoa L.; Dobbins, Roy; Pearsall, Dianna M.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Health Research Center P. O. Box 85122 San Diego, CA 92186-5122			8. PERFORMING ORGANIZATION Report No. 92-25	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Medical Research and Development Command National Naval Medical Center Building 1, Tower 2 Bethesda, MD 20889-5044			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This technical manual contains the information on the program code, data elements, file structures needed to maintain the Computer Assisted Medical Diagnosis System. This documentation was created using the FOXDOC Version 2.0 program.				
14. SUBJECT TERMS Computer Assisted Medical Diagnosis Expert system Technical Medical Automated System			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	